E-Books That Cut It

xun's Guide to EPUB Creation

Copyleft and Thanks

“From all of me to all of you...”

Many thanks to **dearleuk**, who undertook to read early versions of the book and made plenty of helpful comments, all incorporated in the final product.

This is an original EPUB written by xun in December 2019-January 2020.

All text and all images (screenshots and cover) created by xun.

Introduction

EPUB creation is, if not an art, at least a craft. Anybody can load a PDF file into Calibre, click to have it converted to EPUB, and thus be the proud owner of a PDF-to-EPUB conversion.

However, the result is not likely to be something to be proud of. It is likely to resemble a train wreck. Lines that are broken into new paragraphs in the middle of sentences, page numbers left in the middle of the new pages, blocking the reading flow, OCR mistakes that make paragraphs and whole pages impossible to read… This goes for every quick and easy PDF to EPUB converter I have ever seen.

This is also why I would recommend you to download PDF when possible from sources of free and/or public domain books, and then create your own EPUB. Often (but not always, do check!), the EPUBS on offer are precisely like that: converted, but not quality proofed.

If you want to create a really good EPUB from a PDF file, you will have to be prepared to spend some time doing it, and you need to learn to use the tools for doing it properly. It's worth it. You can be truly proud of the result when your work is closing in on perfection…

In other words: use **Sigil**.

Book View (or PageEdit) and Code View

I almost always work in Code View, since I like being in full control of the result, creating style classes as needed. But you can make perfectly decent EPUBs without "coding", too. It is very much better to design in Book View than just stopping after the conversion from PDF to EPUB; that is likely to be a book that is difficult to read due to lots of OCR mistakes, lack of structure and inconsistent formatting.

Working in Book View is faster and easier than being perfectionistic in Code View. That suggests another occasion when Book View may be the better alternative: downloaded books that look awful often have horrid coding and stylesheets under the hood. Instead of looking at the mess and despair, just edit so that the books look OK in Book View!

Since some steps are the same, and other steps are done differently, I have created separate chapters on working in Book View and working in Code View, so that it should be easy to follow the detailed guide without having to skip text that doesn’t apply to your chosen method. This, however, also means that some text is the same in those two chapters, since some steps are indeed the same. If you read both chapters, there will be some *déjà vu* moments.

The Goal

What do you want from an e-book?

You want it to be **legible, easy to read**, so there should no (or a minimum of) misspellings or OCR mistakes turning words into rubbish, and no bad line breaks.

You want to be **able to navigate** the book, so there should be headings and a Table of Content that links to the different parts of the book.

You want **indications when the scene changes**, by a blank line or some special characters, alerting you to move elsewhere in the perspective, space or time of the book world.

You want to be able to **tell utterances from running text**, so there should be quotation marks, correctly and consistently applied, making it clear who says what.

You want to be able to **tell songs, poems, letter, telegrams, and quotations from running text and ordinary utterances**, so these should be formatted somewhat differently (often by a larger left margin and maybe italics).

Your EPUB design should meet those expectations.

I would say that it is also **desirable that the EPUB is designed with different preferences and needs in mind.** Many e-readers allow the reader to choose fonts, adjust font-size and line-height, and even font-weight, and to set left and right margins. Don’t block those choices by creating a stylesheet that forces the reader to use certain fonts, line-heights, etc. To be sure, it is not only a matter of respecting others’ preferences: e-readers are saviours for many people with poor eye-sight, exactly because they can adjust font-sizes, font-weights, etc. and keep reading when paper books are no longer a possibility. Don’t block it by CSS!

In addition, I usually strive to create as clean markup as possible, and a stylesheet that is practically self-explanatory, so that anybody who wants to edit one of “my” books will find it easy to do so. This addition has little to do with the above expectations, though. An EPUB can have horribly confusing markup and unintelligible CSS, and still meet those expectations.

In other words, as long as you meet the reasonable expectations above, you’re fine. Clean markup and clear CSS are nice, but extras.

The Goal of This Book

Ironically, this EPUB is designed to be read at a computer, preferably a desktop computer with a decently large monitor. It doesn’t look that good on a small e-reader. This is a guide, not entertainment. ;-)

Open the book in Calibre’s e-reader on your computer and follow the steps while you work with your book. Select, copy, and paste text from the book as needed.

Using Calibre’s e-reader has the additional advantage of showing the levelled ToC correctly: some e-readers will flatten ToC automatically and show all the little subheadings along with the major headings. Considering the fact that some text is duplicated in the "Working in Sigil (Code View)" and the "Working with Sigil GUI (Book View or PageEdit)" chapters, that provides plenty of room for confusion.

Also, in my opinion, the e-reader’s default font size is a bit largish and so needlessly reduces the amount of text on each page. If you agree, go to Preferences - General tab, and change it.

EPUB2 vs EPUB3

I still do EPUB2. I do know that there are many advantages with EPUB3, but most of them do not matter to the kind of books (novels) I usually convert. I have noticed that even with EPUB2 (HTML4 with CSS2), there are many tags/classes that do not work as they should in every reader. I also know that with HTML5 (EPUB3) we are back to having to check what works in various web browsers, so I suspect that it is quite difficult to predict how tags will work in various readers. Simple, very simple, HTML4/EPUB2 should do.

Disclaimer

This book was mainly written (with a lot of text copied from my old EPUB web page and some of my forum posts at various sites) using an old version of Sigil, 0.8.6, installed on an offline Windows machine. I have checked against later versions (0.9.14 and 1.0.0), but may have missed some changed settings and features, and what's more: these may change in future releases.

In other words, I cannot guarantee that you will not have to look for some settings, similar to the ones I mention, elsewhere in the programs - now or in the future.

Finally…

…this is all about novels, with few or no images and no need for specific layout of pages. If you scan a textbook it may well be best to leave it in PDF, possibly using ScanTailor and other software to fix imperfections. A different kind of work, that.

Workflow Overview

These are brief notes of my workflow when converting books from PDF to EPUB. Some entries are specific for Book View (**BV**) or Code View (**CV**).

The steps are described in more detail in the chapters “Working with Sigil GUI (Book View or Page Edit)” and “Working in Sigil (Code View)”, respectively.

All the time

After scanning, or after opening a PDF you got from elsewhere, and after the conversion to EPUB, make sure that the PDF is OCRed, and thus searchable. Keep it open for quick reference while working on the EPUB.

Images

Copy the cover and save the image.

Also copy the title page if it looks nice enough to insert as a fullpage image instead of making a plain text title page.

In addition, copy other images, if there are any, such as maps or chapter illustrations.

Designing the EPUB in Sigil

1. Remove the text of existing stylesheet (BV and CV) and add your custom CSS (CV). (N.B. check italics first.)
2. Add cover (can of course really be done at any time).
3. Search and replace all &nbsp; with ordinary spaces.
4. Remove HTML ToC. Generate a true EPUB ToC instead.
5. Zoom out the PDF for easy scrolling and scroll through the PDF and EPUB side-by-side as it were.
    - Split the introductory pages, so that titlepage, halftitle, copyright, etc. each get their own page.
    - Optional, for easy overview: rename those pages to halftitle, dedication, etc.
    - Design those special pages one by one.

   - Check if there are special pages at the end of the book, too. If so, split and design as above.
   - Split larger HTML files resulting from the conversion as needed: one HTML file for each chapter, with proper <h> tags.
   - Design any deviant content (letters, telegrams, poems, songs, etc.) in the text.
   - Indicate scenechanges.
   - Add images where they should be.
6. Generate ToC and check that all chapters are there.
7. Optional (CV): add “firstlines”.
8. Optional (CV): replace <p> with <p class="text">.
9. Corrections such as:
   - Spellcheck.
   - Smarten punctuation if needed.
   - Check for faulty line breaks.
   - Check for missing line breaks in conversations.
   - Check for common OCR mistakes that a spellcheck won’t catch.
10. Optional (BV): add your custom CSS.
11. Remove unused style sheet classes.
12. Check with FlightCrew.
13. Proofread.
14. Check with e-reader(s).

Software

For all that I am otherwise on Linux and the occasional Mac, I use Windows for creating EPUBs, in order to have access to my purchased and excellent software. You can run Sigil on Linux and Mac, too, and you can use GIMP or other software for editing images instead of Adobe Photoshop. However, as far as I know, there is nothing as good as ABBYY (and the back-up Nuance Power PDF) for scanning and OCRing PDFs on Linux or Mac.

There is of course a lot of software out there. This is just a catalogue of programs that are well-known in EPUB-making circles, and that I personally have found useful.

I don’t provide links to software here; URLs change. Search the net for those that interest you, and make sure that you download from homepages only. That way there is little risk that your download will contain malware.

PDF Software

Adobe Acrobat, the classic, is fine as long as you have nice, clean PDF files to OCR. If there is a coloured background or somewhat blurry text, Acrobat is not that good in my experience. When I started out, using Acrobat, I used to clean imperfect PDF files as images in Photoshop and then re-assemble them. You will save time using better OCR software if you work with old books and/or imperfect scans. Acrobat XI was the last version for which you could buy a license for continued use of that version; from 2015 onwards, it is part of the “Document Cloud”.

ABBYY is much loved by e-book makers for its great OCR capacities. I use the PDF Transformer+ since I like to be able to do a bit more than just OCR. There is also FineReader, that I believe has a few more options, but use the same OCR engine as PDF Transformer+. ABBYY doesn’t come cheap, but will occasionally be on sale, so it’s worth checking their homepage for price cuts.

Nuance Power PDF. An alternative to ABBYY. They can both cope with PDFs in which the text from the reverse side shines though and the background is brownish. Sometimes one will perform a decent OCR when the other fails. Power PDF uses the same OCR engine as the renowned OmniPage: if you are just going to OCR, or if you work with textbooks rather than novels, you might want to use OmniPage instead, though I have (1) found that I do not need those advanced processing modes, and (2) I like to have a PDF program so that I can convert image PDFs to searchable PDFs.

There is one big annoyance with Power PDF, though: it comes with a poorly integrated Software Manager causing nag screens about updates to appear, and you simply cannot turn them off even if you do not want that particular update. In order to get rid of the software manager, you have to download an uninstaller and also (before you start Power PDF again) remove the FlexNet folder that resides in the hidden Program Data folder.

Like ABBYY, Power PDF is relatively expensive, but is also on sale now and then.

Scan Tailor is free software for editing scans (images); it can be downloaded for Windows and is available in many Linux repositories. For Mac and for Linux distros lacking Scan Tailor in the repos, there is source code to compile. It can be used for fixing orientation, split two-page scans into single pages, and to deskew pages that were skewed during scanning — which often happens when scanning an open book, and tends to cause many OCR mistakes. There is a very good user guide that helps you through the stages of processing the images that will later become a fine-looking PDF.

Briss is a freeware tool for Linux, Mac, and Windows that also allows you to split two-page scans/PDFs. I don’t recommend trying to make an EPUB out of anything but single page PDFs.

Word processing

Atlantis is a word processor for Windows. It looks like an older MS Word version but it is really quite powerful, and has the additional advantage, compared to MS Word, that you can export your documents to EPUB format. It is not freeware, but the license isn’t that expensive, either.

LibreOffice is a freeware Office suite with Linux, Mac and Windows versions; documents created or edited in Writer can be saved directly to EPUB. The resulting markup of the HTML pages is not a pretty sight, though. Personally, I would rather save a document to PDF and then use ABBYY for the EPUB conversion. If you do use LibreOffice (Writer) for exporting to EPUB, do not use the “Clear all direct formatting” option. It removes the all-important italics!

Image editors

Adobe Photoshop (for Mac and Windows) is the classic, and very useful once you learn how to use it. Since 2013 it's subscription software ("Creative Cloud") rather than a license for a version, which means that you can buy the right to use it for a short period of time, but on the other hand you have to keep paying to keep using it.

GIMP, GNU Image Manipulation Program, have many of the Photoshop features, and is free software for Linux, Mac and Windows. Like Photoshop there is a bit of a learning curve, but it is great when you know how to use it.

IrfanView, freeware for Windows is excellent for changing (reducing) the size of an image. You can do some edits in there, too, but the GUI is far from intuitive.

paint.net (note that this is the name of the program, while the URL is getpaint.net) is freeware for Windows. It is probably more intuitive to use if you want a GUI that is easier to start using right away, compared to Photoshop and GIMP, but also not as rich in features.

E-book software

For editing: **Sigil**. I almost only work in the very useful Code View, but it is very nice to be able to quickly change into 'Book View' and see what your code looks like when it has been parsed. Sigil also contains tools that, for example, allows you to create and edit the Table of Contents and check that your EPUB complies with the standard. In addition there is spellchecking (you may need to "train" it) and both normal and Regex search-and-replace, which is really a must. Sigil is downloadable freeware for Mac and Windows, and later versions are included in many Linux repositories. It can also be built from source.

There are many plugins for Sigil. **FlightCrew**, for validating your EPUB, and **SmartenPunctuation** are the most essential ones in my opinion.

N.B. Since version 0.9.2, you cannot "prettify" and tidy HTML code without having it automatically mended. This is not a problem if you "only" work in Book View (since Sigil is unlikely to introduce markup errors that way), but, unless the mending algorithm is perfect (which is unlikely), automatically mending custom HTML code may lead to unwanted changes. Sigil 0.9.2 onwards also mends automatically when you save the EPUB (File - Save or Ctrl+S), but there is still an error message if you try to change from Code View to Book View and there's an error in the markup. Previously (assuming settings like the ones detailed below), there was an error message when trying to save the problem line was somewhat indicated when trying to switch to Book View ("an error on line 160 or above"). From 0.9.2 the page will be rendered in Book View up to the error, with a message saying so, so you can go back to Code View and fix it without having it automatically mended.

In Sigil 0.9.1 and earlier, you can set your preferences so that your HTML code will be “prettified” but not automatically mended. *Set Preferences → “Clear Source”: “Use Pretty Print Tidy”, which will make your HTML code easy to survey. Do not use “HTML Tidy”, since this might perform corrections to your code that may prove fatal for your project, depending on the perceived problem. Check “Automatically Clean and Format HTML Source Code” for “Save”. If there is something syntactically wrong with your code, you well get an error message offering to take care of the problem for you as you save your e-book. I always refuse the offer in order to go find and sort out the problem myself. That way I know the corrections made are the ones I want.*

N.B. Since version 0.9.15, Sigil no longer has the switch to Book View feature, but there is a preview for checking what edits will look like when reading the book. The buttons for quick formatting are still there, but you see the resulting code, not what it looks like in a book.

Instead of the old Book View GUI, there is now a separate program, **PageEdit**, for making GUI edits. It can be launched via Sigil, or used on its own for editing HTML files; you can’t open an EPUB in PageEdit on its own, but again, it can be launched via Sigil, and you can save changes you make in PageEdit and then continue in Sigil (where you need to save the changes again)

If you prefer Sigil with the old built-in Book View, you can still download an old version (0.9.14 or older) and keep using that.

Calibre is a well-known freeware e-book manager for Linux, Mac and Windows that allows you to add tags and other meta data to your books, and it offers a useful overview once you have the proper tags in place.

Calibre is great for converting between the common reflowable text formats AZW, MOBI, and EPUB. You can even use it for creating acceptable (though not really great) PDFs from EPUB. It’s converting from the fixed format PDF to the reflowable EPUB (or AZW or MOBI) that will end with disaster with Calibre and other automatic converters.

Calibre comes with an editor as well. I haven’t used it much (just for quick fixes), always returning to Sigil.

MobileRead forums is where to discuss the making of e-books, and meet the developers for Sigil and Calibre among others. Lots of plugins are developed and maintained there, so it’s worth a look if you feel the need of expanded features.

E-book reader software

When creating EPUBs, it is always good to try and check that your books will work on other devices than the one you currently use. Here are some alternatives for use on your computer.

Adobe Digital Editions can be run on Mac and Windows, and is often required for reading library e-books (and is thus supported by many e-book readers, too).

Sumatra PDF (Windows) can actually handle plenty of formats, including EPUB, and runs with a small footprint.

If you use **Calibre**, there's an e-reader for checking books that can be run on its own, too. It opens all the common formats.

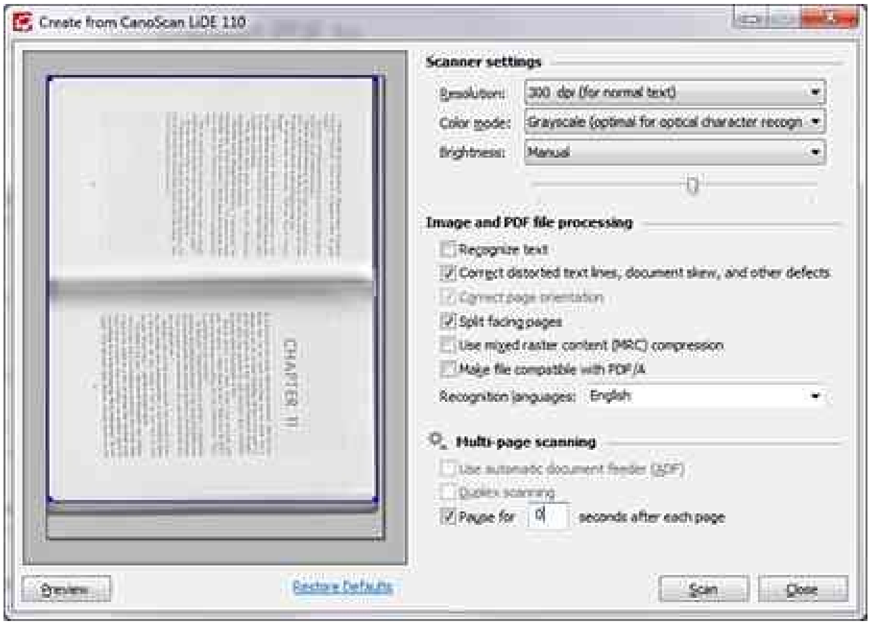
If you convert to AZW or MOBI but don't have a Kindle reader, you can check the result in **Kindle4PC** or **Kindle4Mac**, both free to download from Amazon.

Scanning and Converting to EPUB

Scanning with ABBYY Transformer+

I have a plain little flatbed scanner, and it works well enough for the relatively few books I scan. I scan from ABBY.

These are the settings I usually use; the zero amount of time for turning pages is not so small as it may seem, since the scanner takes its time returning to original position and “warming up” before starting over.



Scanner settings

Resolution: 300 dpi (for normal text)
Color mode: Grayscale (optimal for optical character recogn
Brightness: Manual

Image and PDF file processing

□ Recognize text
☑ Correct distorted text lines, document skew, and other defects
☑ Correct page orientation
☑ Split facing pages
□ Use mixed raster content (MRC) compression
□ Make file compatible with PDF/A
Recognition languages: English

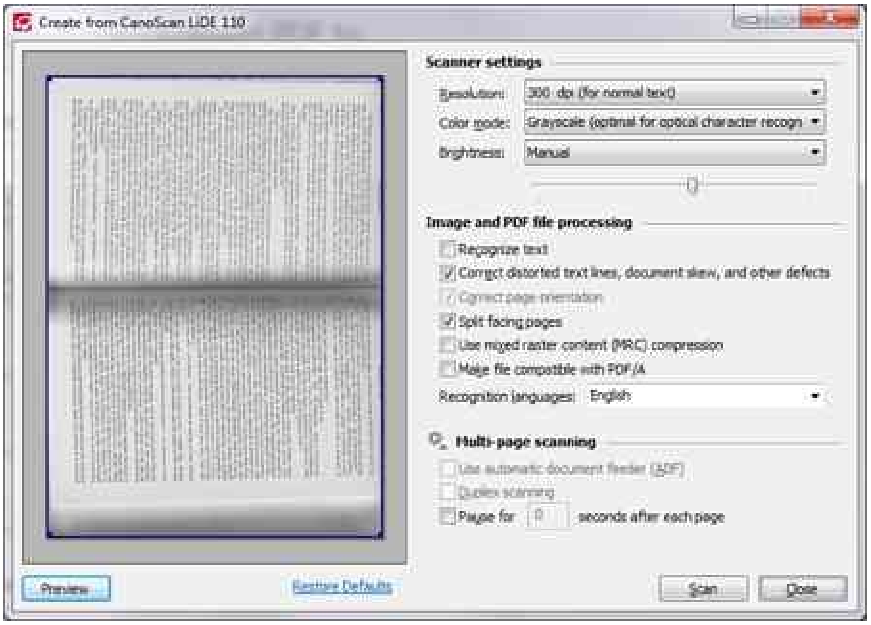
Multi-page scanning

□ Use automatic document feeder (ADF)
□ Duplex scanning
☑ Pause for 0 seconds after each page

Here it doesn’t matter if I turn text recognition on or off while scanning, but I would like to point out that the formatting is lost if I use the similar-looking “OCR” option with the original scanner software. You do not want to lose italics when scanning novels.

You can cancel scanning at any time, and continue with the same document when convenient, and also re-scan any page that for some reason failed to live up to quality expectations. (This is likely to happen if you happen to move the book during the scan.) ABBYY will allow you to choose where to insert new pages.

If possible, I take care not to harm the paper books while scanning. However, some books are fairly hopeless cases. In a case like the one below, you will either have to spend a lot of time correcting misinterpreted words at the start and finish of each line, or slaughter the poor book so that you can scan the pages one by one.



Scanner settings

Resolution: 300 dpi (for normal text)

Color mode: Grayscale (optimal for optical character recogn

Brightness: Manual

Image and PDF file processing

□ Recognize text

☑ Correct distorted text lines, document skew, and other defects

☑ Correct page orientation

☑ Split facing pages

□ Use mixed raster content (MRC) compression

□ Make file compatible with PDF/A

Recognition languages: English

Multi-page scanning

□ Use automatic document feeder (ADF)

□ Duplex scanning

□ Pause for 0 seconds after each page

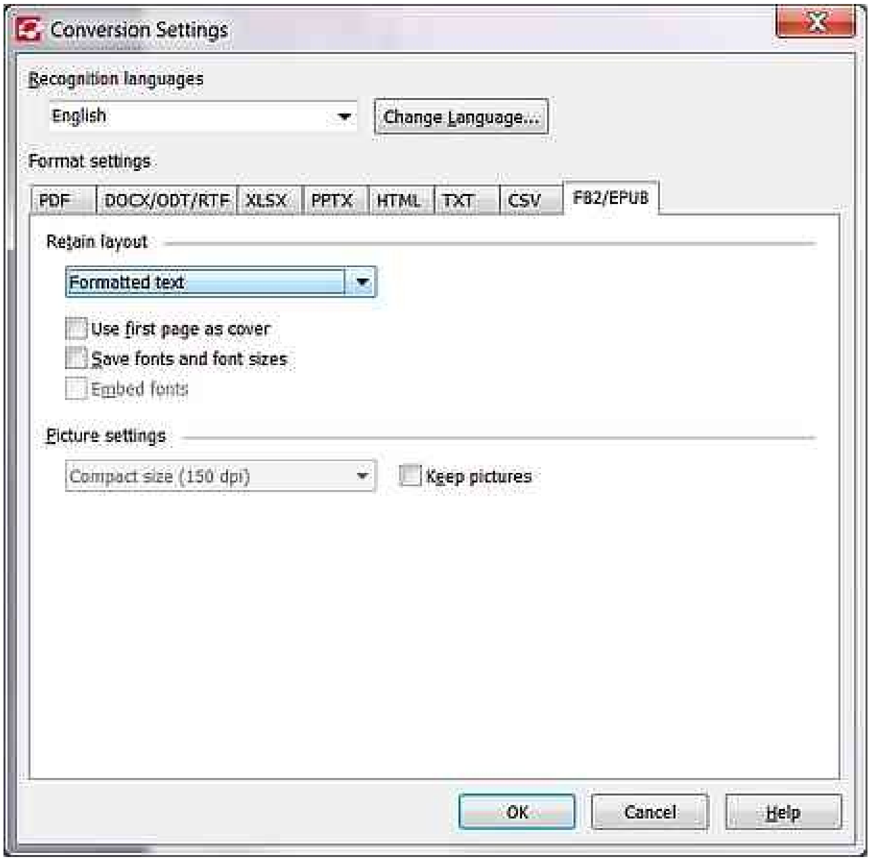
Converting to EPUB

These are my recommended settings for converting from ABBYY to EPUB. (The settings are found in the Convert To menu.)

Selecting “Formatted text” is important. If you select “Plain text” the italics will disappear!

Since ABBYY doesn’t handle images all that well, I don’t even try to make it use the first page as cover, even if there is a cover in the PDF file. It will be added later on.

I also don’t recommend saving fonts and font sizes (that gives you a complicated markup that just takes time to clear up), and absolutely not to embed fonts. Embedded fonts make the EPUB unnecessarily large, and impossible to customize for the reader to boot.



Conversion Settings

Recognition languages: English

Format settings

Retain layout: Formatted text

□ Use first page as cover

□ Save fonts and font sizes

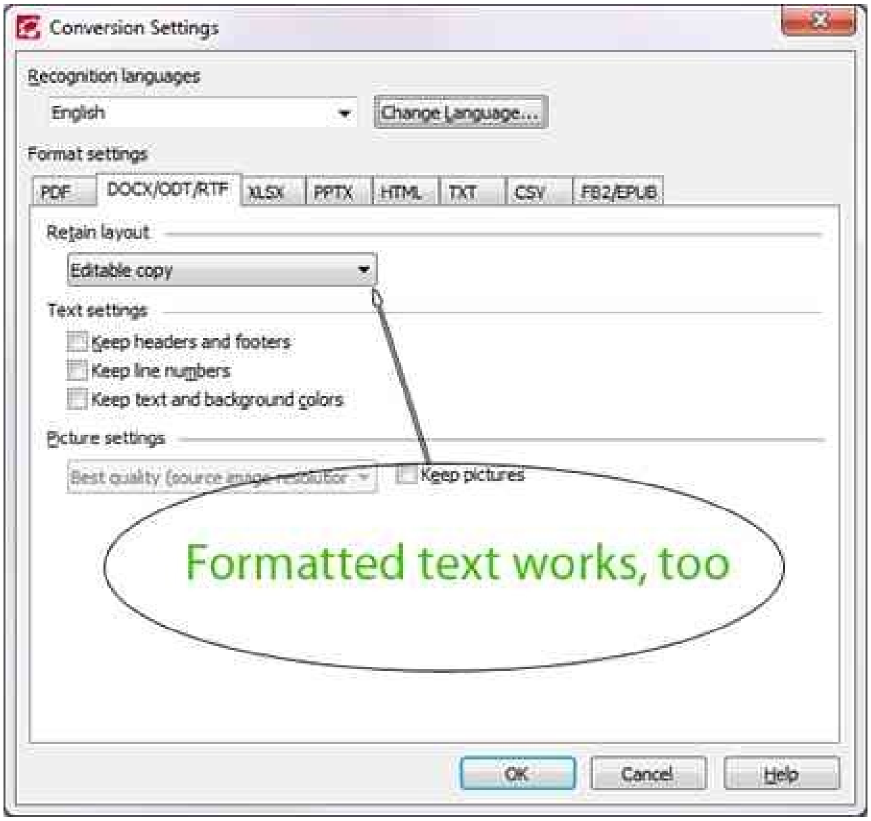
□ Embed fonts

Picture settings: Compact size (150 dpi) □ Keep pictures

Converting to RTF

If there is a lot of reconstructive work to be performed on a bad scan, I will occasionally choose to convert to RTF instead, since I find it easier to re-write lots of text in a real word processor.

These are the settings I use in those rare cases:



Recognition languages: English

Format settings

Retain layout: Editable copy

Text settings

□ Keep headers and footers

□ Keep line numbers

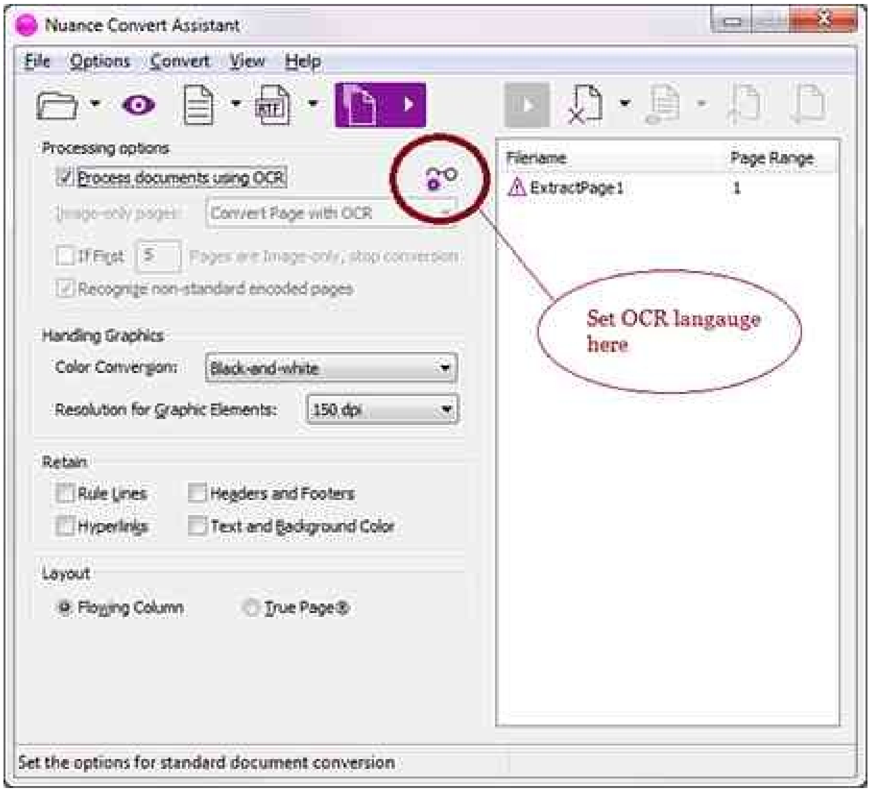
□ Keep text and background colors

Picture settings: Best quality (source image resolution) □ Keep pictures

Formatted text works, too

Timetable or a tabbed list: re-scan page(s) with Power PDF

Mysteries, especially, sometimes contain timetables or tabbed list, illustrating the detective’s attempt to put happenings in order. If any of these have turned out very messy with ABBYY, and you have access to Nuance Power PDF, it may be a good idea to convert those pages using it; my old version can’t convert directly to EPUB, but I save to RTF and then convert from Atlantis to EPUB (a separate one, and then I copy from that one to the book I’m working on). If Nuance Power PDF has managed better, as it often does, it still saves a lot of work.



Processing options

☑ Process documents using OCR

Image-only pages: Convert Page with OCR

□ If First 5 Pages are Image-only, stop conversion

☑ Recognize non-standard encoded pages

Handling Graphics

Color Conversion: Black-and-white

Resolution for Graphic Elements: 150 dpi

Retain

□ Rule Lines □ Headers and Footers

□ Hyperlinks □ Text and Background Color

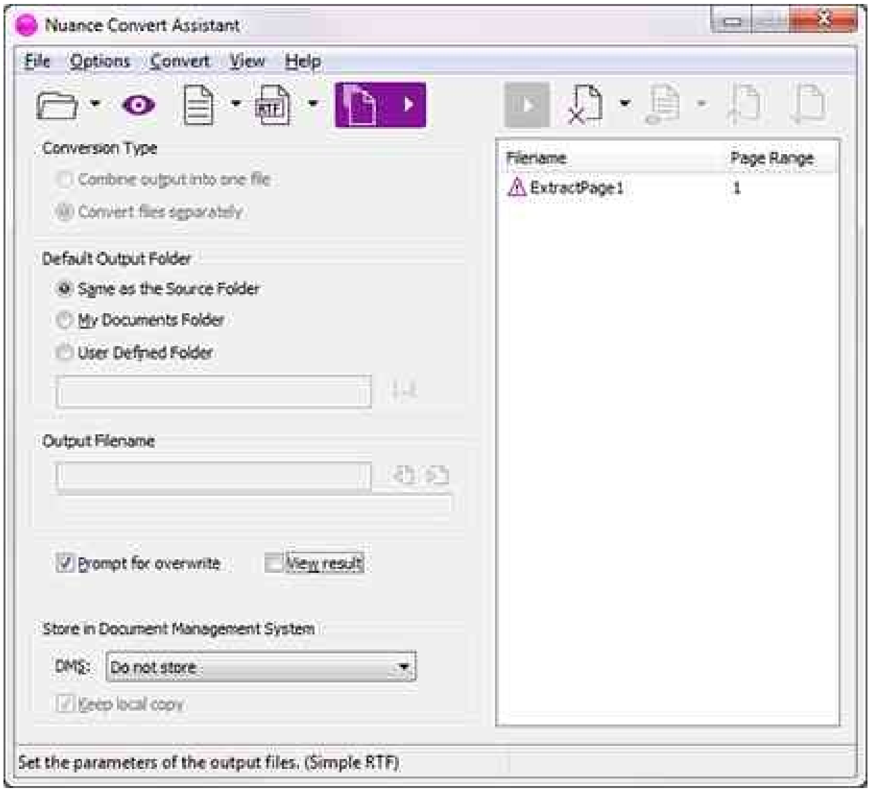
Layout

● Flowing Column ○ True Page®

Filename	Page Range
⚠ ExtractPage1	1

Set OCR langauge here

Set the options for standard document conversion



Conversion Type

○ Combine output into one file

● Convert files separately

Default Output Folder

● Same as the Source Folder

○ My Documents Folder

○ User Defined Folder

Output Filename

☑ Prompt for overwrite □ View result

Store in Document Management System

DMS: Do not store

☑ Keep local copy

Filename	Page Range
⚠ ExtractPage1	1

Set the parameters of the output files. (Simple RTF)

Editing in RTF Before Converting to EPUB, Using Atlantis

Occasionally, a scan will be so poor that the OCR will fail dramatically. This typically happens when you scan a book you can’t press open enough, and not cut into separate pages.

If it is just a matter of skewed lines, it is sometimes possible to make the PDF much better using Scan Tailor. But sometimes the text near the middle will be smudged due to its distance to the scanner, and that is difficult or impossible to recover by processing the PDF.

It’s a matter of taste, but in such cases, when there is a lot of text reconstruction to do, I prefer to do this in a word processor. So, in such rare cases, I don’t convert from ABBYY to EPUB, but instead to RTF.

Never convert to TXT!!! There are still web pages out there that recommend using text files for making EPUBs. It’s madness. A plain text file will lose all formatting, notably the all-important italicized words. Italics convey meaning. Don’t ever lose them!

So, RTF it is: Rich Text Formatting.

I open the RTF for editing in Atlantis in these instances, and export to EPUB when done.

Exporting a document with all formatting intact will result in rather horrid markup, though. I clean out practically everything (except italics!) before exporting to EPUB. That way there are still some, but not that many, annoyances to clear away before the real work with the EPUB can begin.

Autocorrect Settings

Before you open the file with Atlantis, check the autocorrect settings (Tools → Autocorrect Options). I uncheck most of them, but highly recommend the “smart” (curly) quotes setting. Make sure that “Capitalize first letter of sentences” is unchecked — if it is checked, there will be upper case letters in the middle of sentences that were broken by section breaks.

Autocorrect

After opening the document, run “Autocorrect…”. It will give you suggestions for what to correct, so you can reject anything that doesn’t seem like a good idea. It will only do one kind of correction at the time, so you will need to run it again to get all corrections done. With my settings, it only needs to be run once or twice.

Spellchecking

Next; spellchecking, which you might do here and/or later on in Sigil.

Spellchecker options are found at Tools → Options → Spellcheck tab.

You set the language in Spellcheckers & Dictionaries; if you work with both British and American English books, it’s a good idea to first check that you are using the right dictionary.

Check:

- Spellcheck as you type
- Underline misspellings
- Ignore Internet and file addresses

And leave everything else unchecked.

Next, run the spellchecker. Open the searchable PDF and run it alongside Atlantis as you spellcheck. (I usually use Power PDF for this; I prefer the search feature that “only” finds and selects the next occurrence.) If you are unsure about a word that the spellchecker “thinks” is wrong it is very useful to search for it in the original PDF. If it is such a common word that you risk getting too many results, search for some less common word that appears near the one you need to look up.

You may have to cancel the spellchecking and perform a manual correction, for example if a word has been incorrectly hyphenated, like “ob-“ than continues on the next line with “serve”. (The spellchecker may suggest replacements but offers no way to remove the hyphen.) Start the spellchecker where you are done by hitting the F7 key.

How to handle actual misspellings in the original text? Well, that is up to your conscience and your definition of “perfect”… ;-)

Remove Formatting

The last step before converting to EPUB is to remove as much unnecessary formatting as possible. I basically only keep italics (important!) and smallcaps (if applicable). Everything else will be restored by HTML and CSS later on; for now, I want a nice clean CSS to start with.

Atlantis tends to create different classes in the style sheet for very minor perceived differences in the text, and also creates rather horrid HTML markup in which <span> tags, inline elements, are used for whole blocks (that of course should have classes for block tags, usually <p> tags in a book). In other words, removing formatting before conversion saves a lot of time later.

In Font Format - Font: Make sure that the Italic and Small caps are filled, so that these styles will carry over. Font name and size doesn’t matter (since you will noit save and embed fonts). Set the rest to Auto or None.

In Font Format - Spacing, set Scale to 100% and the rest Normal. No kerning.

In Paragraph Format, set Indents and Intervals to 0 (zero) or none, set Alignments to Left. Uncheck everything in the Line & Page Breaks tab except Do not hyphenate.

In File - Page Settings, set margins to any size all around, header and footer to 0)zero). Choose any Page Format (A4 or letter) in the next tab, set orientation to Portrait. Uncheck everything but Supress endnotes in the Layout Tab.

Saving as EPUB

The last step before the fun begins for real: saving the document as an EPUB. Go to File - Save Special - Save as eBook… You can fill in meta data such as Book title and Author if you like, or leave it for later, but one thing is important here: do NOT embed fonts. Select “Don’t save” fonts.

Stylesheets

You have to decide which is your stylesheet philosophy: what to decide for others and what to leave be for reader settings. I have drawn exactly that line: none of my stylesheets will enforce features that can usually be adjusted using e-reader settings.

However, books that I have designed with my “extended” stylesheet will suffer if the CSS file is replaced with the “Replacement CSS”. They will still be quite readable, but poems, songs, and letters will look the same as running text when their classes are gone from the CSS.

It is never easy. Create a book that looks very nice, but is still reasonably customizable, or a book with very basic, barely utilitarian design?

Oh well. There are three stylesheet versions below: a very basic one, an expanded one that contains classes that may come in handy., and an extremely simple one to use as a replacement in books with painfully bad (inadjustable) formatting, or with a GUI-designed book (that will have lots of its styling, such as italics and margins (“indents”) in the HTML files.

Basic Stylesheet (CV):

```
@page {
margin: 5pt
}
/*
body{
margin-top;1em;
margin-right;1em;
margin-bottom;1em;
margin-left;1em;
font-family:Georgia,serif;
font-size:100%;
line-height:1.5em;
color:black;
background-color:white;
widows:1;
orphans:1;
}
*/
/* ======= Standard Elements ======= */
h1{
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
}
h3{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
}
img{
max-height: 100%;
max-width: 100%;
}
/* uncomment p below if you prefer to have the text justified by CSS */
/*
p{
text-align:justify;
}
*/
/* ======= Inline Style Classes ======= */
.bold{
font-weight:bold;
}
.italic{
font-style:italic;
}
/* ======= Block Style Classes ======= */
.centered{
text-indent:0;
text-align:center;
}
.right{
text-align:right;
}
.small{
font-size:0.75em;
}
/* ======= Block Content Classes ======= */
.firstline{
text-indent:0;
margin:1em 0 0 0;
}
```

```
.scenechange{
text-indent:0;
margin:1em 0 0 0;
}
.text{
text-indent:1.2em;
margin:0;
}
```

Expanded Stylesheet (CV)

```
@page {
margin: 5pt
}
/*
body{
margin-top;1em;
margin-right;1em;
margin-bottom;1em;
margin-left;1em;
font-family:Georgia,serif;
font-size:100%;
line-height:1.5em;
color:black;
background-color:white;
widows:1;
orphans:1;
}
*/
/* ======= Standard Elements ======= */
h1{
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
}
h3{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
}
img{
```

```
max-height: 100%;
max-width: 100%;
}
/* uncomment p below if you prefer to have the text justified by CSS */
/*
p{
text-align:justify;
}
*/
/* ======= Inline Style Classes ======= */
.bold{
font-weight:bold;
}
.firstletter{
font-size:1.5em;
}
.italic{
font-style:italic;
}
.quotefix{
word-spacing:-0.1em;
}
.ref {
vertical-align:super;
font-size:1.2em;
text-decoration:underline;
}
.smallcaps{
font-variant:small-caps;
}
/* ======= Block Style Classes ======= */
.centered{
text-indent:0;
text-align:center;
}
.left{
text-indent:0;
text-align:left;
margin:0;
}
.noindent{
margin:0;
text-indent:0;
}
.right{
text-align:right;
}
.small{
font-size:0.8em;
}
/* smallcaps are used when left at the beginning of chapters, where it doesn't really matter if they are parsed
*/
```

```
.smallcaps{
font-variant:small-caps;
}
.xmargin{
margin:0 0 0 2em;
text-indent:0;
}
/* ======= Block Content Classes ======= */
.alsoby{
text-indent:0;
font-size:0.8em;
margin:0;
}
.author{
text-align:center;
font-size:1.2em;
font-weight:bold;
margin:1em 0 0 0;
}
.booklist{
text-indent:0;
font-size:0.8em;
margin:0;
}
.copyright{
text-indent:0;
margin:0;
text-align:center;
font-size:0.6em;
}
.copyright-fl{
text-indent:0;
margin:1em 0 0 0;
text-align:center;
font-size:0.6em;
}
.dedication{
text-align:center;
text-indent:0;
font-style:italic;
margin:1em 0 0 0;
}
.disclaimer{
text-indent:0;
font-style:italic;
margin:3em 0 0 0;
}
.firstline{
text-indent:0;
margin:1em 0 0 0;
}
.footnote{
```

```
text-indent:0;
font-size:0.7em;
margin:0.7em 0 0.7em 0;
}
.halftitle{
text-align:center;
text-indent:0;
font-size:1.2em;
margin:3em 0 0 0;
}
.letter{
margin:0 0 0 1.5em;
text-indent:1.2em;
}
/* poem class can usually be used for songs, too */
.poem{
margin:0 0 0 1.5em;
text-indent:0;
}
.publisher{
text-indent:0;
text-align:center;
margin:3em 0 0 0;
}
.scenechange{
text-indent:0;
margin:1em 0 0 0;
}
.text{
text-indent:1.2em;
margin:0;
}
.title{
font-weight:bold;
font-size:1.4em;
text-align:center;
margin:1em 0 0 0;
}
```

Replacement CSS (BV & CV)

Assuming that you have an EPUB with <h> tags for headings and <p> tags for paragraphs, you may choose this stylesheet (edited to your preferences, or copied as it is) for books “only” edited in GUI, too. That way you can have paragraphs with indents and no margins. Remember to make blank lines for scenechanges if you “only” edit in GUI!

I have added classes for bold, italic and two possible varieties of scenechange, too. They do no harm if they are not used in the text, but will save boldness, italics and scenechanges if they *are* in use.

```
h1{
text-indent:0;
text-align:center;
margin:1em 0 0.5em 0;
font-size:1.5em;
font-weight:bold;
}
h2{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1.2em;
font-weight:bold;
}
h3{
text-indent:0;
text-align:center;
margin:1em 0 0.3em 0;
font-size:1em;
font-weight:bold;
}
img{
max-height: 100%;
max-width: 100%;
}
/* uncomment "text-align:justify;" below if you prefer to have the text justified by CSS */
p{
text-indent:1.2em;
margin:0;
/*text-align:justify;*/
}
.bold{
font-weight:bold;
}
.italic{
font-style:italic;
}
.scenebreak{
text-indent:0;
margin:1em 0 0 0;
}
.scenechange{
text-indent:0;
margin:1em 0 0 0;
}
```

Comments on the Stylesheet Classes

When I started making EPUBs from pdfs, I tried to imitate what the books looked like in paper versions. It was often quite fun trying to come up with solutions that didn't depend on unreliable things like pseudoclasses and dropletter tags.

As should be clear by now, I changed my mind, and advocate creating books that are easy to customize using e-reader features such as choosing fonts, font size and font weight, line-height, and margins. The stylesheet settings are deliberately simple in order to make sure that they work for the majority of e-readers.

Classes in stylesheets and HTML files

Apart from the standard elements, such as "h2", "img", and "p", I have added a number of classes to the stylesheets. You see that they are classes by the dot before their names, such as .firstline and .text. When a stylesheet entry is defined as a class, it can be used many times in the same HTML page (unlike an ID, that has to be unique).

A stylesheet entry has no effect if there is nothing in the HTML files that corresponds to the entry. If there are no <h2> or <h3> tags in the HTML files, the h2 and h3 settings in the stylesheet will have nothing to affect.

You make a stylesheet class entry like .firstline take effect by adding it inside a tag in a HTML file. By changing <p> to <p class="firstline">, you add the styling and formatting described for ".firstline" in the stylesheet to that paragraph, which will make it look different from paragraphs using <p class="text"> or <p class="copyright">, etc.

Inline classes are used inside paragraphs, notably in <span> tags, such as <span class="italic">*some words*</span>, and affect the words between the opening and closing tags only.

Comment out

You can make comments in CSS (as well as in HTML and most other codes). A commented stylesheet line (or text block) starts with /* and ends with */. A commented line may contain settings, but they have will have no effect, unless/until the line is uncommented. See, for example, the “body” entry in my stylesheets: I leave it commented out, but if somebody who gets one of my books wants to specify margins, background colour, etc. for the whole book, they can uncomment that part and edit the settings.

Uncomment

A commented line (or text block) starts with /* and ends with */. Remove those, and the line (such as “text-align:justify;”) and the line is uncommented and goes live, making the settings affect the HTML element it is applied to.

@page

Small margin that will apply to all pages, set in fixed size pt since a relative unit would make the margins very large if the font size is much increased by the reader.

.author, .publisher and .title

...are used on the title page. Manual replacement needed, of course, but it’s just the three paragraphs.

body

The body element is commented out in order to ensure maximum freedom for the reader to specify margin, font, line-height, colours, etc. It’s there as a service to those who might want to edit the settings.

.booklist

Used when there’s a page with “also by” or similar. Replace <p> with <p class="booklist"> on current page (after splitting).

.copyright

Replace <p> with <p class="copyright"> on current page. This assumes you have split pages, of course, so that copyright is on its own page. I insert -fl into the copyright tags manually (changing them to <p class="copyright-fl"> where there’s a blank line above a line. Not necessary, but looks nicer.

.img

```
img{
max-height: 100%;
max-width: 100%;
}
```

...will make sure that images don’t overflow, and thus don’t go too large for the reader

.firstletter

Simply making the first letter in the first line beginning a chapter larger than normal text; in the expanded stylesheet it’s 1.5 em.

.firstline

...is not really necessary, but IMHO it is nice to use it for the start of chapters (and after scenechanges if this are done by inserting a <p>&nbsp;</p> line).

.footnote

The text of the footnote, at the bottom of the HTML page.

p

The stylesheets contain a pargraph setting for justified text that you only need to uncomment if you want justified text and your reader doesn’t support making it so. The setting is commented out for accessibility reasons: justified text may be hard do read if you need very large fonts, and the irregular spaces between words may also make the text harder to read for people with dyslexia. Many e-readers allow you to set text alignment according to your preference - but this will only work if “text-align” isn’t specified in the stylesheet (or by style in the HTML page).

.quotefix

This is used for single quotes directly after double quotes, or vice versa. It is very far from necessary; I use it because it’s neat and easy to apply with a normal search; it “ties” single and double quotes together with a &nbsp; while reducing the distance between them a little, so that a spoken line starting or ending with somebody quoting somebody else looks good. Happens a lot in mysteries! There are instructions for applying it in the Corrections chapter.

N.B. this is a simple formatting work-around. The most elegant solution would be not to use spaces between them, but increase the spacing so that they don’t run into each other. Unfortunately it seems that some e-readers will not parse spacing settings, so work-around is it. If the “quotefix” isn’t parsed by an e-reader, the non-breaking space will still prevent a single quote from ending up at a new line.

.ref

Used for the link (usually a digit) to a footnote. Underlined superscript, somewhat larger than the normal font so that it’s easier to click.

.smallcaps

The class is used when leaving smallcaps in, for example at the start of chapters.

The small-caps font variation is unreliable: not all e-readers will parse it. I usually leave them be if they are in the start of chapters; it does no harm if they aren’t parsed. I change to upper case with a small span class for signs and similar, that needs to be in “caps” no matter if the reader can parse smallcaps or not. Note that there is a button (“AB”) in Sigil for quickly changing all letters in a selection to upper case, so it is quick and easy work.

.text

...is the class I use for everything that could be left plain <p>. In most cases it is not necessary, but there are some readers that won’t parse CSS settings for plain <p>, and so they will be stuck with blank spaces between paragraphs no matter the margin:0 setting in the style sheet. To use, just replace <p> with <p class="text"> on all HTML pages when all other styling is done.

Working with Sigil GUI (Book View or PageEdit)

You don’t *have to* work in Code View. You have more control over the final product if you apply your knowledge of HTML and CSS in Code View, but using the GUI will go a long way. Especially compared to just converting with ABBY (or Calibre!) and leaving it like that.

One problem with the GUI is that you can’t, for example, apply the zero margin and the indent you might like to use for most of your paragraphs (<p>) in a novel. There will be spaces (like empty lines) between all your paragraphs, and there will be no text-indent in paragraphs.

You also can’t change all the headings in a way that suits you. You can certainly click and add italics, or centering, etc., but all you get is tags for the individual clicked instance. You have to click all your h2 headings, for instance, if you want them to look a certain way.

In short: you can’t specify styles when using the GUI. That doesn’t have to be a problem. Just go with the defaults, and you’ll be fine…

Or, provided that your EPUB contains ordinary <h> and <p> tags, use the “Replacement CSS” from the Stylesheets chapter.

However, using Sigil’s GUI with a stylesheet with your specific formatting decisions will sometimes lead to unwanted changes in the design. Sigil reads the stylesheet settings and may apply them as styles in the text when you remove lines or move parts of the text. Paragraphs may get styles that then cannot be edited without going into Code View in order to see just what happened.

You may end up with a heading going larger than the other ones of the same level because Sigil added a 1.2em span style to the existing 1.2em font-size in the style-sheet, or a heading getting a paragraph class when moved up by deleting previous lines, or indeed a heading changed into a paragraph with the heading attributes set in the shape of span styles; that (previous) heading will not be included when you generate a new ToC.

You can add styles at three locations: in the text of the HTML file (along with the HTML tags), at the top of the HTML page, and in an external stylesheet (a CSS file) that the HTML page links to. Styles in the text of the HTML page override styles at the top of the page as well as styles in a stylesheet/CSS file. (Styles at the top of the page override styles in a stylesheet/CSS, but not those in the text.) So: styles in the text of the HTML file can mess up what you want to accomplish with a stylesheet.

This is why the workflows for Code View and Book View are different when it comes to adding a stylesheet. **When working in Book View, only add the stylesheet when all corrections are done**, assuming that you don’t want blank spaces (margins) between your paragraphs, and also want a text-indent for those same paragraphs (which makes novels nicer to read). You can correct misspelled words without getting unexpected style changes, so there is no problem with adding the stylesheet before proofreading.

Of course, you can temporarily add the stylesheet before that, in order to check what the book will look like. But do remove it before you continue editing.

Save a copy

There is an option to save a copy in the Files menu. Use it before making thorough changes, such as using Replace All when correcting problems. Such a copy may prove very valuable if the result turns out to be undesirable, but you noticed too late.

Sigil plugins

There are a lot of plugins for Sigil, but in my opinion only two that are essential:

Flight Crew is a plugin for validating your EPUB, making sure that it will work as intended in the vast majority of e-readers. If there are errors, there is also information about where they are and what kind of error it is. Most of the time, there will of course be no errors.

Smarten punctuation turns straight quotes into curly quotes as needed, and will do a lot for making the novel feel nice to read.

Searching

Go to Search → “Find & replace…” in order to open the searchbox (at the bottom of the window).

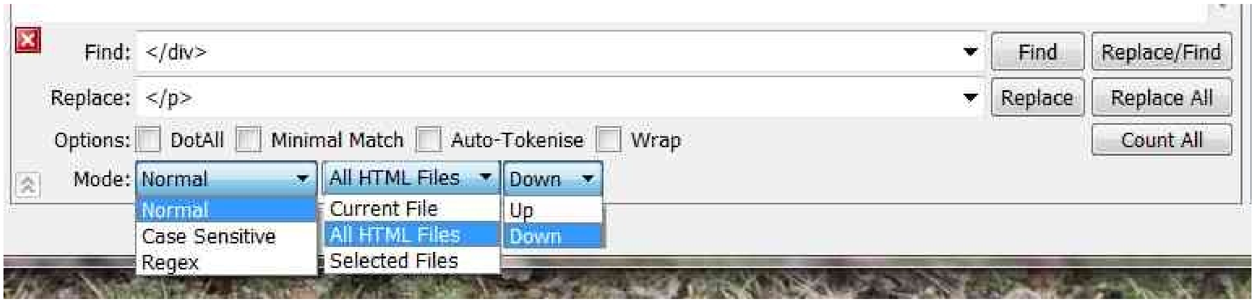
When you want to search the CSS or apply replacements to one page only, make sure that you have selected **Current File**. When you want t search the whole book, select **All HTML Files**.

When you search with **Regex**, it simply will not work unless you have set the search mode to Regex. Normal searches will usually work in Regex, too, unless your search terms happen to include any of the “special characters” that have specific meanings in Regex. Best to switch to Normal mode for normal searches!

Normal mode (or **Case Sensitive**) is “only” for the text as you see it in Code View.

If you stay with the default **Down** direction of searches, make sure that you start your search at the top of the page when searching Current File, and at the very start of the book when you search All HTML Files.

This is an illustrative fake image: you can’t have all the selections showing at once without faking it:



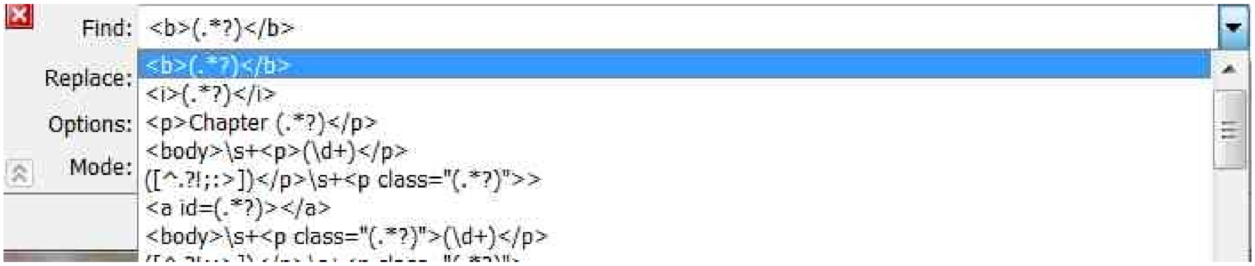
Find: </div>

Replace: </p>

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Normal, All HTML Files, Down

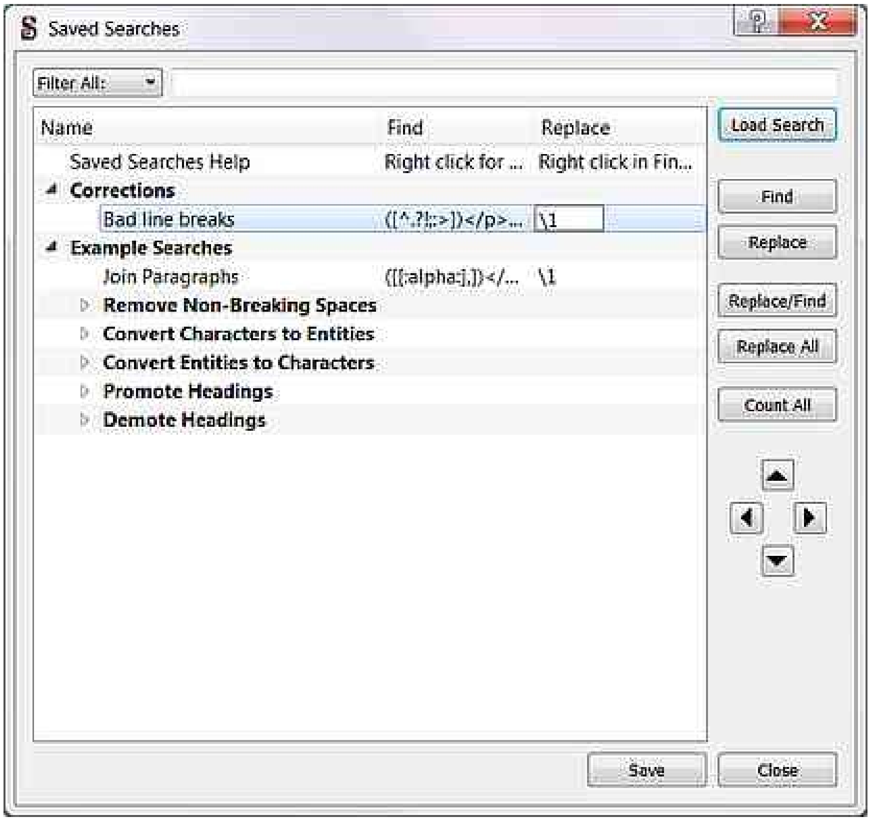
The latest searches are saved in reverse order of appearance (latest saved searches first). This is of course very useful for searches that you repeat many times: just click the little down arrow to the right of the text field (either “Find” or “Replace”) and select the search term you want to use.



Find: <b>(.*?)</b>

```
<b>(.*?)</b>
<i>(.*?)</i>
<p>Chapter (.*?)</p>
<body>\s+<p>(\d+)</p>
([^.?!;:>])</p>\s+<p class="(.*?)">>
<a id=(.*?)></a>
<body>\s+<p class="(.*?)">(\d+)</p>
```

You can also add searches by going to Tools → “Saved Searches...”. There are a number of pre-defined searches there already. You can add and delete Groups and Entries by right-clicking in the window. Here I have added a “Corrections” group, and a search for bad line breaks:



Name	Find	Replace
Saved Searches Help	Right click for ...	Right click in Fin...
Corrections		
Bad line breaks	([^.?!;:>])</p>...	\1
Example Searches		
Join Paragraphs	([[:alpha:],])</...	\1
Remove Non-Breaking Spaces		
Convert Characters to Entities		
Convert Entities to Characters		
Promote Headings		
Demote Headings		

Saving useful text to paste

There is a **Clip Editor** under “Tools” where you can store often-used phrases (which is perhaps most useful when working in Code View, but deserves a mention here, too). Right-click to add Groups and/or Entries, the same way as in Saved Searches. You can access your clips (and useful example ones) with a right-click in the text you are working on.

Detailed workflow

This workflow will be described with Book View in mind. It is basically the same if you use the free-standing PageEdit. Note that you have to open your EPUB with Sigil even if you use PageEdit for GUI editing, and that you can only work on one HTML file at the time in PageEdit. In my opinion, it is best to stay with an older Sigil version if you prefer to work in Book View.

What follows is instructions for what to do if you work according to the previously outlined workflow. This chapter “only” deals with design, though. The last parts of the workflow, corrections, have got their own chapters.

Images

With the PDF open in ABBYY (or other PDF software), copy the cover and save the image in Paint (or other program that allows you to paste and then save in bmp, tif, or possibly as png). Open it and fix as necessary in Photoshop, GIMP, or other similar software. Crop the cover if needed, change brightness/contrast if needed, do other fixes as needed, then save in the original format (such as png, bmp or tif). If the cover is in too poor shape, leave it as it is or go hunting the webs for a better image.

Open the cover image in IrfanView and resize to something suitable – I usually downsize to a height about 750 pixels. IrfanView is much better for downscaling with decent quality than Photoshop or GIMP.

Save in jpg if the image is photolike, with shades and nuances, to gif if there are are clearly contrasted full-colour fields. The quality can be low, about 30 of 100 or so; this takes the file size down.

Also copy the title page if it looks nice enough to insert as a fullpage image instead of making a plain text title page.

In addition, copy other images, if there are any, such as maps or small illustrations. For black-and-white images (typically titlepage and maps), convert to greyscale (from colour), then use brightness/contrast and/or bucketfill the background with white. Save with no other changes; again use Irfanview for downscaling and converting to gif (or jpg if photolike). The size of images that are smaller than fullpage will of course depend on the context; it's a bit of trial and error, so make sure that you keep the original.

Now: open your EPUB in Sigil. Note that you can change between Book View and Code View by clicking the open book and <> buttons above.

Adding HTML files

Sometimes (rarely), you will need to add HTML pages in Sigil, for example in order to insert a fullpage map into the middle of the book.

Right-click the HTML file above the desired place of the new page, and choose "Add blank HTML file". It will be called "section-something" and there's no need to rename it. It is the order of files, as shown in the left-hand book browser, that is important.

The new file will have a very basic head section. Notably, the link to the CSS file will not be copied, and the result is of course that your CSS (if you add a stylesheet) will have no effect there. So, go to an “old” page in Code View and copy the head section of the HTML file. Typically it will look something like this:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<link href="../Styles/main.css" rel="stylesheet" type="text/css" />
</head>
```

Paste it into the new page (in Code View), replacing the old head section. See “Fullpage images (large maps, copied title page, etc.)” for how to insert such images.

Remove the text in any existing stylesheet

As mentioned above, stylesheet settings can be applied in unwanted ways when editing in Book View. So, open the Styles folder and double-click the CSS file, then remove all text. If you do want to apply styles by stylesheet, such as having text-indent for paragraphs in a novel, add the stylesheet when all the editing and all the corrections are finished. The step is included in the last chapter (“The Finishing Touches”).

However, if you are not working with an EPUB created by exporting from ABBYY, but one created some other way (or downloaded as EPUB), you have to check the stylesheet first. **If the italics in the book depend on that stylesheet, removing it will cause a small disaster.**

In such cases, you need to search the stylesheet for “italic” settings, and leave those classes in the stylesheet, now and when you paste your own one when the book is finished. (Personally, I would search for the classes that have “italic” in them in the HTML files, remove the span styles, and instead apply italics using Sigil. And then remove those classes from the stylesheet, too.)

Add cover

Go to Tools → “Add cover…” and browse to your saved (jpg or gif) image. Sigil will create a new cover file, placing it first in the list of HTML files.

Search and replace all &nbsp; with ordinary spaces

ABBYY makes lots of them. IMHO it’s OK to use &nbsp; as needed while designing, but these are superfluous, so get rid of them first. (If you have exported from some other software, or are working on somebody else’s EPUB, you may need to search for &#160; instead.)

If the search box is not open, go to the Search menu and click “Find and replace…”. Use normal mode and search all HTML files. Assuming that the default direction “Down” is used, place your cursor at the very first line of the book before you start.

Find:	Replace with:

&nbsp;
or
&#160;
	a blank space — it is very important not to forget this, since no space here means all words running together…

Index

If there is an index and the OCR has left it too messy, I simply delete it. (N.B. Nuance often performs better than ABBYY on indexes, lists and tables, so if you have access to it it might be worth OCRing such pages with it, if ABBYY has made a mess.) If it looks OK or needs very little editing, I leave it, but make no effort to make it “live” by linking. I leave it in order to show what the author found worth indexing.

Remove HTML ToC and generate ToC

Look for a table of contents with links and remove it. Search for <ul> if you can’t spot it and it’s not on its own page. If it is on its own page, simply right-click the HTML file and delete the page.

Generate a ToC instead, either by clicking the icon somewhat top right, or by going to Tools → “Table of Contents” → “Generate Table of Contents”. Do this right away (and then again when the book is finished); often ABBYY will have added h-tags that don’t show in the ToC when the book is first opened after conversion.

Main book design

Zoom out the PDF for easy scrolling and **scroll through the PDF and EPUB**

side-by-side as it were.

Splitting HTML files

If an introductory page has not got its own HTML file (as it ideally should), you can easily split the file(s): place the cursor at the right place, then hit Ctrl+Enter.

(You can also place markers and then perform numerous splits in one go. Place the cursor at the right place, then go to Insert → "Split Marker". When all the markers are in place, go to Edit → "Split At Markers".)

Don't worry if the names of the new HTML files are not pretty. The names don't matter — though I personally usually rename the special pages for easy overview.

Special pages design

The first pages of the book may be titlepage, halftitle (just the title of the book, while "titlepage" usually has author name and publisher name, too), copyright page, maybe credits and a dedication. Apply bold and italic styles as needed (compare to the PDF) by selecting text and clicking the buttons above. If there is a source for an epigraph or a quote, it usually looks best if you align that paragraph to the right.

Create HTML files in the EPUB as needed for each chapter

Split larger files resulting from the conversion. Search in the EPUB for an unusual word near the chapter beginning in order to quickly find it.

Split the same way as you did for the introductory "special" pages: place the cursor at the right place, then hit Ctrl+Enter. Don't worry if the names of the new HTML files are not pretty. The names don't matter.

An EPUB will work even if you don't bother to have one HTML page per chapter (and one HTML page per special page), but apart from being neat, it also makes conversions to MOBI or AZW better, since they tend to "look" at HTML files.

Optional (if you are orderly, like me:) if there are orphaned HTML files due to splitting at chapters, copy and paste the text into the correct chapter HTML file, *then delete the superfluous files. One XHTML file for each chapter.*

Make the chapter heading (number and/or name) h2 or h3.

Select text for a heading, then click one of the h1-h6 buttons. Perfect markup, much easier than adding <h2> (etc.) tags manually.

Special styles (letters, poems…)

If and when you spot any deviant items, such as **telegrams, letters, poems, and songs**, style them so that they stand out from the running text, similar to but not necessarily the exact same way as they do in the PDF.

Most of the time this will mean adding blank lines above and below the letter (etc.), increasing the left margin (the “indent” button with arrow to the right), and applying italics in some cases.

Smallcaps

If there are SMALLCAPS, just leave them be if they are at the start of chapters. They will not be parsed by all e-readers, but that will do no harm there. If there are smallcaps in the running text, for signs or telegrams or newspaper headlines, etc., select the words that should be in caps and click the “AB” button somewhat to the right above. That will turn all the letters into upper case.

Indicate scenechanges

…whenever you see a scenechange indicated in the PDF. Indicate scenechanges by adding a blank line or a new line with some visual clue such as ***. Center the asterisks by clicking the centering button above.

A scenechange (aka scenebreak) is when, for example, a new paragraph takes place in Paris instead of London. Most books have a blank line or some little image, or one or more asterisks, between such paragraphs, so that the reader will be made aware that there is a “jump” in the text.

More images?

If you spot any **images** that you haven’t already copied, copy them from the PDF, fix and rescale them, so that you can insert them in the EPUB where they should be – and insert any already copied and fixed images you saved earlier.

You insert images by going to Insert → “File…” and then clicking the “Other files…” button. Browse to where you have saved the image to be added. If you have downscaled and saved as jpg or gif as instructed, make sure that you don’t accidentally add the original, needlessly large, one instead.

Fullpage images (large maps, copied title page, etc.)

In order to make fullpage images cover a full page, while keeping correct ratio and not overflow (turn larger than the e-reader can show), you need to go into Code View and apply some changes to a (new) blank HTML page.

Use the Insert feature first, inserting the image in its own otherwise blank HTML page, so that the image is added to the book, then edit the page. If you don’t know the size of the image, open the Images folder in the Book Browser pane to the left and double-click the image; the information is there.

A cover added using the Add Cover tool is marked up like this; I use it for other images that need to be full-page, too, just changing the measurements (two places each, width and height in the “viewBox”, then height and width for the image) and of course the file name. The bold text shows what will need to be changed.

```
<div style="text-align: center; padding: 0pt; margin: 0pt;">
<svg xmlns="http://www.w3.org/2000/svg" height="100%" preserveAspectRatio="xMidYMid meet"
version="1.1" viewBox="0 0 472 750" width="100%" xmlns:xlink="http://www.w3.org/1999/xlink">
<image height="750" width="472" xlink:href="../Images/cover.jpg" /></svg>
</div>
```

Footnotes

Some novels contain footnotes, though usually not as many as there can be in text books.

The easiest way of dealing with them is to place an asterisk where the reference is, then select it and click the A^2 button so that it becomes superscripted. Like this: *. Then place the footnote text it refers to in its own paragraph right below the one with the reference, adding “* [FOOTNOTE]” before the footnote text. No linking involved.

* [FOOTNOTE] So, here is the text to read when you have spotted the * in the paragraph above.

If you prefer to place the footnote texts at the chapter (HTML page) end instead, with linking back and forth, here’s how to do it:

Place the footnote text at the bottom of the HTML page. Add “1” or “*” or whatever you will use as a reference in the text at the start of the paragraph.

Select the “1” or “*” and click the anchor button. You will be asked to “insert ID”. You need to give the anchor you are about to create a unique name, as these anchors are used as targets when linking back and forth. I usually call the anchors at the bottom of the page “back1”, “back2”, etc., since they will later be used to link back to where the footnote reference is in the running text.

Next, type the reference (“1” or “*”) in the text and click the chain button next to the anchor. This will provide you with lots of targets to choose between, because all the headings in the ToC will show up. Scroll to your newly made anchor (“back1” in my case) or use the search feature in order to find it. Select it and then click OK. The reference* now links to the footnote text.

But, you want to be able to go back to the text after reading the footnote. So: select the reference in the text again, but this time click the anchor button. As before, you need to choose a unique name; I usually use “note1”, “note2”, etc.

With that ID in place, scroll down to the bottom of the page and select the “1” or “*” you used as the footnote text anchor. Click the chain button and scroll through available targets; select the one you created for the reference (“note1” in my case). You may want to add a line (just press the ___ key) above the footnote text, making it clear that it is not part of the running text.

Generate ToC again

Do this when all the chapters have got their headings (and their own HTML pages), and then check that all the chapters are there. You can add or remove items in the ToC, and also edit the entries by going to Tools → “Table of Contents” - “Edit Table of Contents...”. This is useful when the chapter headings are so long that the ToC becomes hard to read.

Unexpected and unwanted markup when editing in Book View: <div> instead of <p>

Notably, I have noticed that ABBYY will sometimes create a <div> instead of a

<p> when you add a new line by pressing Enter in Book View. These <div>s will of course not be affected by the style settings in your CSS, so you want to change them to <p>. If you spot it, simply select the <div> paragraph and click the “P” button somewhat to the left above. That should change it all from <div> to <p>. Also run a Normal search for <div>, and change them to <p> the same way.

Spellcheck

Sigil has a built-in spellchecker that is quite powerful, not least because you can add new words to it. In order to add a word to the dictionary, select it and click the “Add To Dictionary” button.

You can also add new replacements, such as “well” for “weU” by editing in the suggestion text box and then clicking “Change Selected Word To:”.

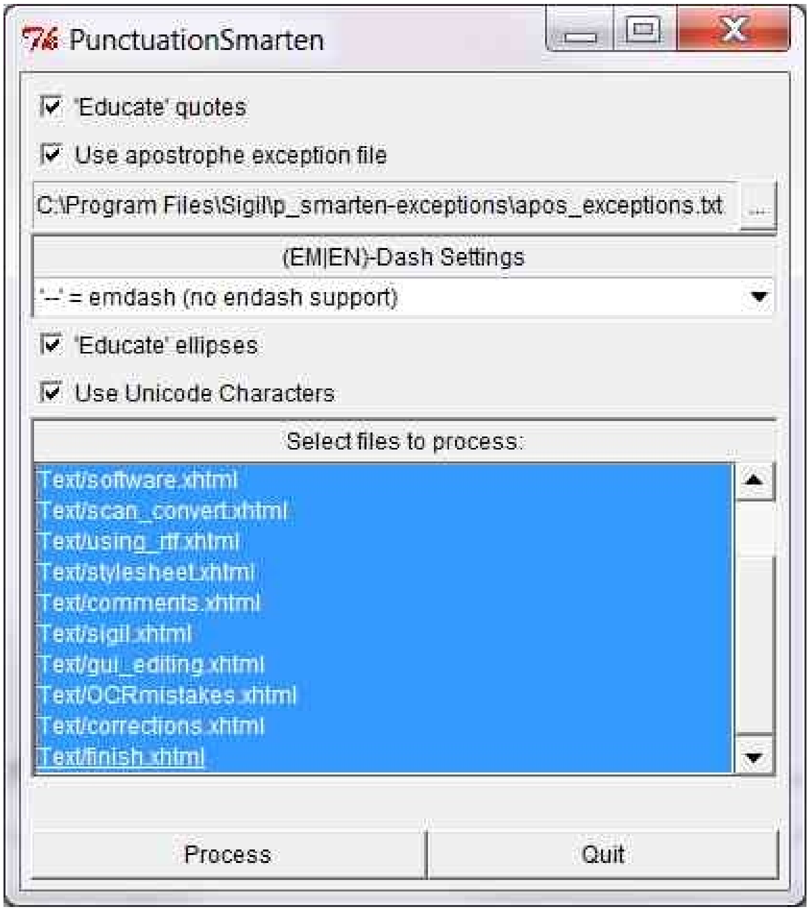
In addition, you can ignore words. This means that the word in question is ignored only for the duration, which is useful for names that may be differently spelled in another book. You don’t want to add a specific spelling to the dictionary. In order to ignore, select the word and click the “Ignore” button.

Apart from the spellchecker, that runs in a little window and lists possibly misspelled words, you can check “Highlight Misspelled Words” in Preferences - Spellcheck Dictionaries. Those words will then be highlighted (underlined with red) in Code View, not Book View, but this still allows you to see them in context, just like you are probably used to from word processors.

Smarten punctuation

Assuming that you have installed the plugin, go to Plugins → “Edit…” → “Punctuation Smarten”, and then click “Start”.

A small windows opens. I leave everything checked there:



PunctuationSmarten

☑ 'Educate' quotes

☑ Use apostrophe exception file

C:\Program Files\Sigil\p_smarten-exceptions\apos_exceptions.txt

(EM|EN)-Dash Settings

'--' = emdash (no endash support)

☑ 'Educate' ellipses

☑ Use Unicode Characters

Select files to process:

Text/software.xhtml
Text/scan_convert.xhtml
Text/using_rtf.xhtml
Text/stylesheet.xhtml
Text/comments.xhtml
Text/sigil.xhtml
Text/gui_editing.xhtml
Text/OCRmistakes.xhtml
Text/corrections.xhtml
Text/finish.xhtml

Select all files.

Then click “Process”. It usually doesn’t take very long to have all quotation marks smartened. Finish by clicking “OK”.

Next: corrections

These will be carried out the same way no matter if you work in Code or Book View. Your EPUB is basically ready now, but it is likely full of OCR mistakes, bad line breaks, and other issues which means that you still have not met those reasonable expectations…

---

* Now there’s linking back and forth.

Working in Sigil (Code View)

Sigil plugins

There are a lot of plugins for Sigil, but in my opinion only two that are essential:

Flight Crew is a plugin for validating your EPUB, making sure that it will work as intended in the vast majority of e-readers. If there are errors, there is also information about where they are and what kind of error it is. Most of the time, there will of course be no errors.

Smarten punctuation turns straight quotes into curly quotes as needed, and will do a lot for making the novel feel nice to read.

Save a copy

There is an option to save a copy in the Files menu. Use it before making thorough changes, such as using Replace All when correcting problems. Such a copy may prove very valuable if the result turns out to be undesirable, but you noticed too late.

Searching

Go to Search → “Find & replace…” in order to open the searchbox (at the bottom of the window).

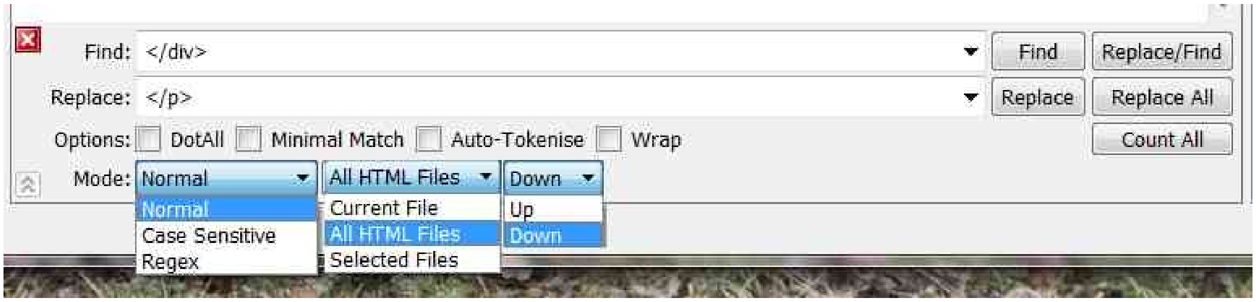
When you want to search the CSS or apply replacements to one page only, make sure that you have selected **Current File**. When you want t search the whole book, select **All HTML Files**.

When you search with **Regex**, it simply will not work unless you have set the search mode to Regex. Normal searches will usually work in Regex, too, unless your search terms happen to include any of the “special characters” that have specific meanings in Regex. Best to switch to Normal mode for normal searches!

Normal mode (or **Case Sensitive**) is “only” for the text as you see it in Code View.

If you stay with the default **Down** direction of searches, make sure that you start your search at the top of the page when searching Current File, and at the very start of the book when you search All HTML Files.

This is an illustrative fake image: you can’t have all the selections showing at once without faking it:



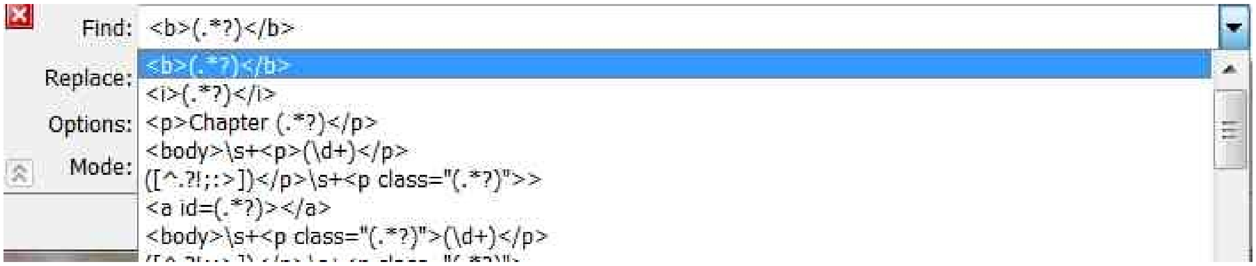
Find: </div>

Replace: </p>

Options: □ DotAll □ Minimal Match □ Auto-Tokenise □ Wrap

Mode: Normal | All HTML Files | Down

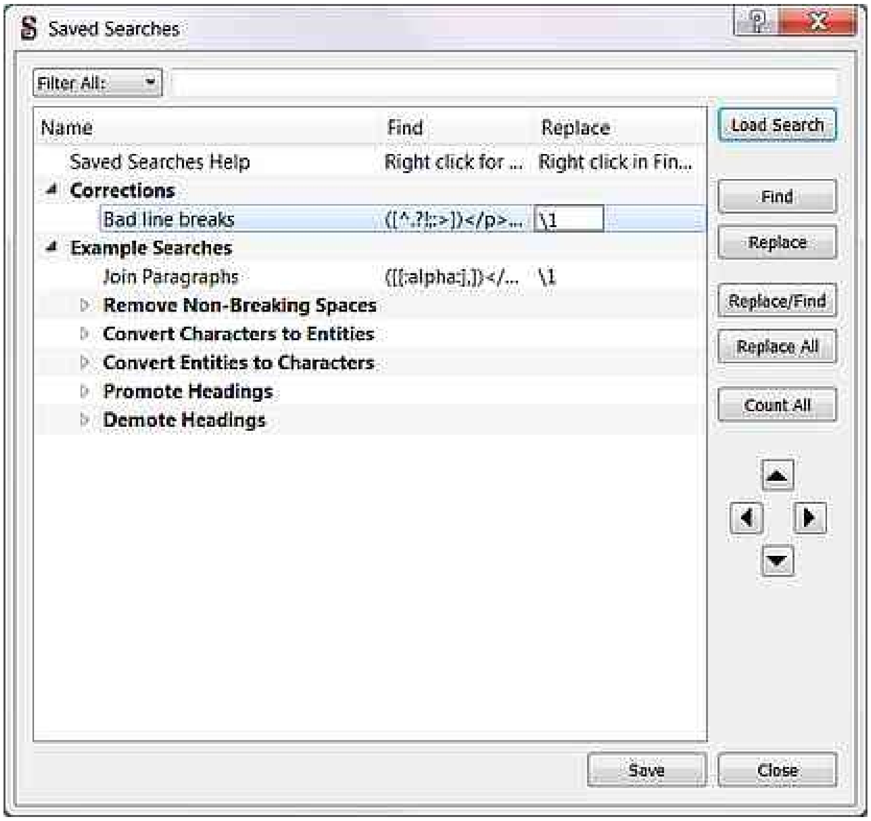
The latest searches are saved in reverse order of appearance (latest saved searches first). This is of course very useful for searches that you repeat many times: just click the little down arrow to the right of the text field (either “Find” or “Replace”) and select the search term you want to use.



Find: <b>(.*?)</b>

```
<b>(.*?)</b>
<i>(.*?)</i>
<p>Chapter (.*?)</p>
<body>\s+<p>(\d+)</p>
([^.?!;:>])</p>\s+<p class="(.*?)">>
<a id=(.*?)></a>
<body>\s+<p class="(.*?)">(\d+)</p>
```

You can also add searches by going to Tools → “Saved Searches...”. There are a number of pre-defined searches there already. You can add and delete Groups and Entries by right-clicking in the window. Here I have added a “Corrections” group, and a search for bad line breaks:



Name	Find	Replace
Saved Searches Help	Right click for ...	Right click in Fin...
Corrections		
Bad line breaks	([^.?!;:>])</p>...	\1
Example Searches		
Join Paragraphs	([[:alpha:],])</...	\1
Remove Non-Breaking Spaces		
Convert Characters to Entities		
Convert Entities to Characters		
Promote Headings		
Demote Headings		

Saving useful text to paste

There is a **Clip Editor** under “Tools” where you can store often-used phrases. Right-click to add Groups and/or Entries, the same way as in Saved Searches. You can access your clips (and useful example ones) with a right-click in the text you are working on.

Detailed workflow

What follows is instructions for what to do if you work according to the previously outlined workflow. This chapter “only” deals with design, though. The last parts of the workflow, corrections, have got their own chapters.

Images

With the PDF open in ABBYY (or other PDF software), copy the cover and save the image in Paint (or other program that allows you to paste and then save in bmp, tif, or possibly as png). Open it and fix as necessary in Photoshop, GIMP, or other similar software. Crop the cover if needed, change brightness/contrast if needed, do other fixes as needed, then save in the original format (such as png, bmp or tif). If the cover is in too poor shape, leave it as it is or go hunting the webs for a better image.

Open the cover image in IrfanView and resize to something suitable – I usually downsize to a height about 750 pixels. IrfanView is much better for downscaling with decent quality than Photoshop or GIMP.

Save in jpg if the image is photolike, with shades and nuances, to gif if there are are clearly contrasted full-colour fields. The quality can be low, about 30 of 100 or so; this takes the file size down.

Also copy the title page if it looks nice enough to insert as a fullpage image instead of making a plain text title page.

In addition, copy other images, if there are any, such as maps or small illustrations. For black-and-white images (typically titlepage and maps), convert to greyscale (from colour), then use brightness/contrast and/or bucketfill the background with white. Save with no other changes; again use Irfanview for downscaling and converting to gif (or jpg if photolike). The size of images that are smaller than fullpage will of course depend on the context; it’s a bit of trial and error, so make sure that you keep the original.

Now: open your EPUB in Sigil. Note that you can change between Book View and Code View by clicking the open book and <> buttons above.

Adding HTML files

Sometimes (rarely), you will need to add HTML pages, for example in order to insert a fullpage map into the middle of the book.

Right-click the HTML file above the desired place of the new page, and choose “Add blank HTML file”. It will be called “section-something” and there’s no need to rename it. It is the order of files, as shown in the left-hand book browser, that is important.

The new file will have a very basic head section. Notably, the link to the CSS file will not be copied, and the result is of course that your CSS has no effect there. So, go to an “old” page and copy the head section of the HTML file. Typically it will look something like this:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
```

```
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<link href="../Styles/main.css" rel="stylesheet" type="text/css" />
</head>
```

Paste it into the new page (in Code View), replacing the old head section. See “Fullpage images (large maps, copied title page, etc.)” for how to insert such images.

Add your custom CSS

Use one from the Stylesheets chapter here, or create your own. My instructions will take my stylesheets for granted, so you have to re-interpret for your own stylesheet, or your edits of mine, when reading this text.

Open the “Styles” folder and double-click “main.css” to open it. Select all the text there and delete it. Copy the text of one of the stylesheets here, and paste it into the css file. Save (File → “Save”, or Ctrl+S, just like any word processor).

The CSS from ABBY holds very little information; the formatting is done in the text. I replace it all with CSS classes, so that practically no styling is done in the HTML pages. See the Corrections chapter for common replacements — these can be done at any time, so no need to rush over there now!

However, if you are not working with an EPUB created by exporting from ABBYY, but one created some other way (or downloaded as EPUB), you have to check the stylesheet first. **If the italics in the book depend on that stylesheet, removing it will cause a small disaster.**

In such cases, you need to search the stylesheet for “italic” settings, and copy those classes to your own stylesheet. (Personally, I would search for the classes that have “italic” in them in the HTML files, remove the span styles, and instead apply italics using my stylesheet class, or add tags by clicking in Sigil. And then remove those classes from the stylesheet.)

Add cover

Go to Tools → “Add cover…” and browse to your saved (jpg or gif) image. Sigil will create a new cover file, placing it first in the list of HTML files.

Search and replace all &nbsp; with ordinary spaces

ABBYY makes lots of them. IMHO it’s OK to use &nbsp; as needed while designing, but these are superfluous, so get rid of them first. (If you have exported from some other software, or are working on somebody else’s EPUB, you may need to search for &#160; instead.)

If the search box is not open, go to the Search menu and click “Find and replace…”. Use normal mode and search all HTML files. Assuming that the default direction “Down” is used, place your cursor at the very first line of the book before you start.

Find:	Replace with:

&nbsp;
or
&#160;
	a blank space — it is very important not to forget this, since no space here means all words running together…

Index

If there is an index ;and the OCR has left it too messy, I simply delete it. (N.B. Nuance often performs better than ABBYY on indexes, lists and tables, so if you have access to it it might be worth OCRing such pages with it, if ABBYY has made a mess.) If it looks OK or needs very little editing, I leave it, but make no effort to make it “live” by linking. I leave it in order to show what the author found worth indexing.

Remove HTML ToC and generate ToC

Look for a table of contents with links and remove it. Search for <ul> if you can’t spot it and it’s not on its own page. If it is on its own page, simply right-click the HTML file and delete the page.

Generate a ToC instead, either by clicking the icon somewhat top right, or by going to Tools → “Table of Contents” → “Generate Table of Contents”. Do this right away (and then again when the book is finished); often ABBYY will have added h-tags that don’t show in the ToC when the book is first opened after conversion.

Main book design

Zoom out the PDF for easy scrolling and **scroll through the PDF and EPUB side-by-side** as it were.

Splitting HTML files

If an introductory page has not got its own HTML file (as it ideally should), you can easily split the file(s): place the cursor at the right place, then hit Ctrl+Enter.

(You can also place markers and then perform numerous splits in one go. Place the cursor at the right place, then go to Insert → “Split Marker”. When all the markers are in place, go to Edit → “Split At Markers”.)

Don’t worry if the names of the new HTML files are not pretty. The names don’t matter — though I personally usually rename the special pages for easy overview.

Special pages design

I don’t try to mimic the paper book exactly, but will use classes for copyright (small), halftitle, author, booktitle, and publisher, and also for dedication, etc. as needed. Check for special pages both at the start and the end of the book. Split HTML files as needed so that each special page has its own HTML file. Optional, for easy overview: rename those pages to halftitle, dedication, etc.

When these are all on their own respective HTML pages, it is easy to find all <p> and replace with the appropriate class, such as <p class="copyright"> for the copyright page. N.B. it is very important to have the Find and replace box set to Current file when performing these replacements. You don’t want the whole book formatted in the smallish copyright style.

In some cases you can’t simply search and replace all <p> tags in the current file. Author, title and publisher for the title page will have to be added manually. I insert “-fl” manually for blank lines at the copyright page — you can simply add a blank line instead. Sometimes there is an epigraph with the source on the line below; I usually set that source line to <p class="right"> manually.

Create HTML files in the EPUB as needed for each chapter

Split larger files resulting from the conversion. Search in the EPUB for an unusual word near the chapter beginning in order to quickly find it.

Split the same way as you did for the introductory “special” pages: place the cursor at the right place, then hit Ctrl+Enter. Don’t worry if the names of the new

HTML files are not pretty. The names don't matter.

An EPUB will work even if you don't bother to have one HTML page per chapter (and one HTML page per special page), but apart from being neat, it also makes conversions to MOBI or AZW better, since they tend to "look" at HTML files.

Optional (if you are orderly, like me:) if there are orphaned HTML files due to splitting at chapters, copy and paste the text into the correct chapter HTML file, then delete the superfluous files. One XHTML file for each chapter.

Make the chapter heading (number and/or name) h2 or h3.

Select text for a heading, then click one of the h1-h6 buttons. Perfect markup, much easier than adding <h2> (etc.) tags manually.

Special styles (letters, poems…)

If and when you spot any deviant items, such as **telegrams, letters, poems, and songs**, style them so that they stand out from the running text, similar to but not necessarily the exact same way as they do in the PDF. The extended stylesheet contains classes for "letter" and "poem", that should also work OK for telegrams and songs, respectively (if you do not wish to add your own new classes).

If you use the buttons for bold or italic text, you will end up with <b>- and <i>tags. It's not a catastrophy even if you prefer to use CSS for all formatting, though. Just replace <b>(.*?)</b> with <span class="bold">\1</span> and <i>(.*?)</i> with <span class="italic">\1</span> using Regex.

Smallcaps

If there are SMALLCAPS, just leave them be if they are at the start of chapters. They will not be parsed by all e-readers, but that will do no harm there. If there are smallcaps in the running text, for signs or telegrams or newspaper headlines, etc., remove the smallcaps span style/class, then select the words that should be in caps and click the "AB" button somewhat to the right above. That will turn all the letters into upper case.

If you are particular like me, you will change those paragraphs to <p class="small">, but it is perfectly fine to just leave the uppercase letters as they *are.*

Add scenechange tags

…whenever you see a scenechange indicated in the PDF; there is a “scenechange” class in all the stylesheets here, so that would be a change from <p> to <p class="scenechange">. This is IMPORTANT: abrupt changes with no indication makes for poor reading experience. Alternatively, indicate scenechanges by adding a blank line (<p>&nbsp;</p>) or some visual clue such as <p class="centered">***</p>.

More images?

If you spot any **images** that you haven’t already copied, copy them from the PDF, fix and rescale them, so that you can insert them in the EPUB where they should be – and insert any already copied and fixed images you saved earlier.

You insert images by going to Insert → “File…” and then clicking the “Other files…” button. Browse to where you have saved the image to be added. If you have downscaled and saved as jpg or gif as instructed, make sure that you don’t accidentally add the original, needlessly large, one instead.

Fullpage images (large maps, copied title page, etc.)

Use the Insert feature first, inserting the image in its own otherwise blank HTML page, so that the image is added to the book, then edit the page. If you don’t know the size of the image, open the Images folder in the Book Browser pane to the left and double-click the image; the information is there.

A cover added using the Add Cover tool is marked up like this; I use it for other images that need to be full-page, too, just changing the measurements (two places each, width and height in the “viewBox”, then height and width for the image) and of course the file name. The bold text shows what will need to be changed.

```
<div style="text-align: center; padding: 0pt; margin: 0pt;">
<svg xmlns="http://www.w3.org/2000/svg" height="100%" preserveAspectRatio="xMidYMid meet"
version="1.1" viewBox="0 0 472 750" width="100%" xmlns:xlink="http://www.w3.org/1999/xlink">
<image height="750" width="472" xlink:href="../Images/cover.jpg" /></svg>
</div>
```

Footnotes

Some novels contain **footnotes**, though usually not as many as there can be in text books. These, that are originally usually placed at the bottom of a paper page, go to the end of the chapter (HTML page), with linking back and forth.

In the text:

```
<span class="ref"><a id="back1" href="#note1">1</a></span>
```

The footnote (placed at the very end of the chapter):

```
<p class="footnote"><a id="note1" href="#back1">1 </a>[FOOTNOTE TEXT]</p>
```

You may want to add a line (just press the ___ key) above the footnote text, making it clear that it is not part of the running text.

Generate ToC again

Do this when all the chapters have got their headings (and their own HTML pages), and then check that all the chapters are there. You can add or remove items in the ToC, and also edit the entries by going to Tools → “Table of Contents” → “Edit Table of Contents...”. This is useful when the chapter headings are so long that the ToC becomes hard to read.

Replace <p> with <p class="text">

Now everything that shouldn’t be running text is tagged differently; replace <p> with <p class="text">. You can leave it <p> and add margin:0 to the style sheet for “p” if you wish, but rumour has it that some e-readers will not parse CSS for classless <p>, so in that case there will be spaces between all paragraphs on those readers.

Spellcheck

Sigil has a built-in spellchecker that is quite powerful, not least because you can add new words to it. In order to add a word to the dictionary, select it and click the “Add To Dictionary” button.

You can also add new replacements, such as “well” for “weU” by editing in the suggestion text box and then clicking “Change Selected Word To:”.

In addition, you can ignore words. This means that the word in question is ignored only for the duration, which is useful for names that may be differently spelled in another book. You don’t want to add a specific spelling to the dictionary. In order to ignore, select the word and click the “Ignore” button.

Apart from the spellchecker, that runs in a little window and lists possibly misspelled words, you can check “Highlight Misspelled Words” in Preferences - Spellcheck Dictionaries. Those words will then be highlighted (underlined with red) in Code View, not Book View, but this still allows you to see them in context, just like you are probably used to from word processors.

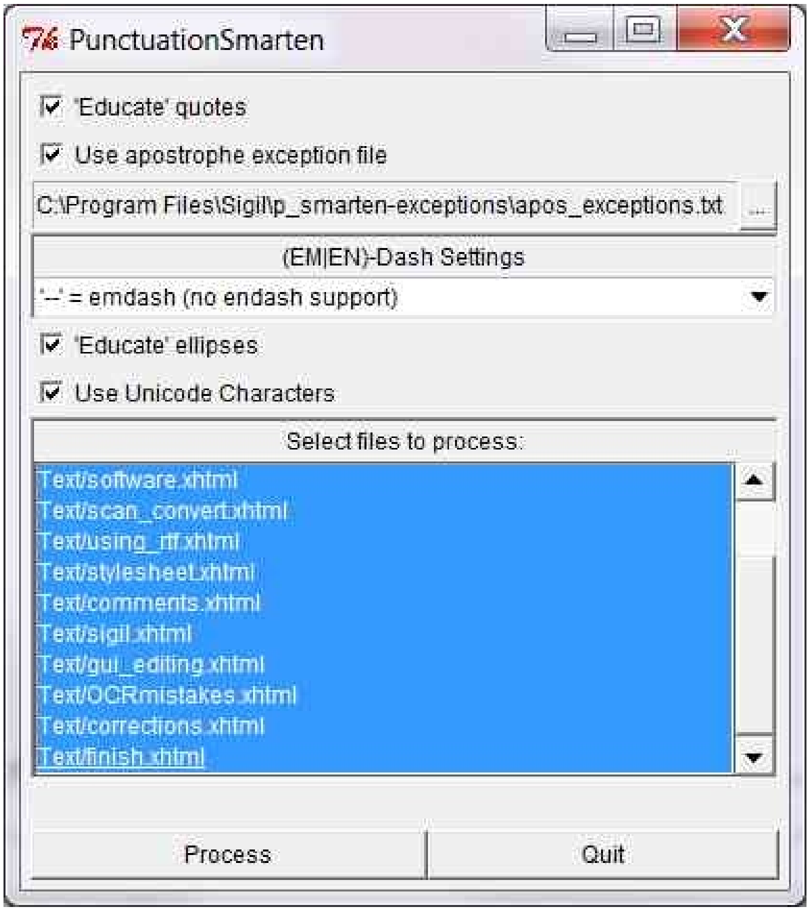
If you prefer to see misspelled words highlighted in something like Book View, you might consider installing the later version of Sigil, and also install the new PageEdit program and “link” it to Sigil. N.B. that the Book View is gone from later Sigil versions, so you will have to do with a small preview pan when not opening a HTML file in PageEdit. (*Personally, I stick to an oldish Sigil version for the main EPUB work, but have the latest version and PageEdit installed on a virtual Windows machine.*)

PageEdit allows you to check spelling the well-known reddish underline way of many word processors, without seeing the underlying code. The drawback is that you can’t “train” the dictionary by adding new words.

Smarten punctuation

Assuming that you have installed the plugin, go to Plugins → “Edit…” → “Punctuation Smarten”, and then click “Start”.

A new small windows opens. I leave everything checked there:



☑ 'Educate' quotes

☑ Use apostrophe exception file

C:\Program Files\Sigil\p_smarten-exceptions\apos_exceptions.txt

(EM|EN)-Dash Settings

'--' = emdash (no endash support)

☑ 'Educate' ellipses

☑ Use Unicode Characters

Select files to process:

```
Text/software.xhtml
Text/scan_convert.xhtml
Text/using_rtf.xhtml
Text/stylesheet.xhtml
Text/comments.xhtml
Text/sigil.xhtml
Text/gui_editing.xhtml
Text/OCRmistakes.xhtml
Text/corrections.xhtml
Text/finish.xhtml
```

Select all files.

Then click “Process”. It usually doesn’t take very long to have all quotation marks smartened. Finish by clicking “OK”.

Next: corrections

These will be carried out the same way no matter if you work in Code or Book View. Your EPUB is basically ready now, but it is likely full of OCR mistakes, bad line breaks, and other issues which means that you still have not met those reasonable expectations…

Corrections and Fixes

Note: “BV” indicates if you should do the search if you work in Book View, while “CV” indicates that you should run the search if you work in Code View. Obviously, some searches should be run in both cases.

Some CV searches, such as adding firstline and firstletter formatting, are clearly optional.

You want to have as many problems as possible corrected by searches before it’s time to proofread your book. It saves a lot of time not having to break off reading and perform manual corrections all the time.

There is an Appendix with most of the searches below in an easier to copy format. (I think I may have forgotten to add some variations detailed below, but I am confident that you will, in that case, be able to add them for yourself.)

If you are creating many EPUBs, it may also be a good idea to add the searches in "Saved Searches", where you can store “Find” and “Replace” together. You can right-click in the Searchbox and save the current search from there, and also load saved searches by right-clicking in the box.

When I occasionally say “run manually” below, it is usually short for: “Click Find, check the result, and if it should be replaced with the option in the Replace box, click Replace (or Replace/Find for immediate search for the next instance), but don’t click Replace All. If the result shouldn’t be changed at all, click Find again. If it should be replaced with something other than what’s in the Replace box, edit it manually (for real).”

Using Regex mode

These are searches that should be run on a newly designed EPUB, but the list is not, and cannot be, complete. There will practically always be some unforeseen problem. You will probably spot it as you proofread and so able to correct it by editing the text, but if you create many EPUBs it is well worth while to learn how to write your own Regex searches in order to deal with repeat problems. N.B. there are different “flavours” of Regex, so if you follow some online course or guide, make sure that it is the right “flavour” by testing the given examples in Sigil.

For missing chapter heading tags (BV & CV)

For cases of <p> instead of <h2> (or <h3>, etc., change to suit you). Of course there shouldn’t be any of these if you have designed the book properly, but you never know... Also, this one is useful if you edit a downloaded EPUB.

To use when there is a pattern (such as “Chapter…”):

Find:	Replace with:
<p>Chapter (.*?)</p>	<h2>Chapter \1</h2>

To use when there are only chapter numbers. This will find chapter numbers only, provided that there is nothing else between the <body> tag and the chapter number. You may have to remove anchors first (see below):

Find:	Replace with:
<body>\s+<p>(\d+)</p> or <body>\s+<p class="(.*?)">(\d+)</p>	<body><h2>\1</h2> or <body><h2>\2</h2>

For page numbers remaining after OCR: (BV & CV)

(This one is useful if you edit a downloaded EPUB, too.)

To use when there is a pattern (such as “Page…”) — beware of “Page…” mentioned in the text, though, so don’t replace all at once.

Find:	Replace with:
<p class="(.*?)">Page \d+</p> or <p>Page \d+</p>	nothing (blank “Replace” box)

To use when there are only page numbers. N.B. if there are chapter numbers in <p> or <p class="text">tags, fix them before doing this (see above).

Find:	Replace with:
<p class="(.*?)">\d+</p> or <p>\d+</p>	nothing (blank “Replace” box)

For bad line breaks (BV & CV)

The first search here is an update (due to a clever question from dearleuk) about searching for paragraphs that begin with a lower case letter instead of just searching for paragraphs that don't end with a proper sentence-closing character. I would recommend first searching for these lower case paragraphs, and would indeed be tempted to replace them all in one go: paragraphs that legitimately start with a lower case letter will be extremely rare. Afterwards, run the second search for paragraphs ending with presumably wrong characters: that one will find missing period marks along with various OCR artefacts, and should still be run "manually". There will be fewer hits when all the lower case lines have already been found, and so the correction work will be faster.

1. Paragraphs that begin with a lower case letter. **N.B. a space before the replacement.**

Find:	Replace with:
</p>\s+<p class="(.*?)">([a-z]\w*\s)	\2
or	or
</p>\s+<p>([a-z]\w*\s)	\1
	N.B. the space before the replacement

2. A catch-all that finds paragraphs that don’t end with period (full stop), question mark, exclamation mark, semi-colon, colon, or the last part of a closing span tag. Check one by one, sometimes there is just a period mark/full stop missing at the end of a paragraph rather than a bad line break, or it could end with em dash… **N.B. replacement followed by a space.**

Find:	Replace with:
([^.?!;:>])</p>\s+<p class="(.*?)">	\1
or	*N.B. the space after “\1”*
([^.?!;:>])</p>\s+<p>	

For missing line breaks in conversations (BV & CV)

For double main quotes. Run manually (since some conversations do not have linebreaks):

Find:	Replace with:
&rdquo;(.)&ldquo;	&rdquo;</p><p class="text">&ldquo;
or	or
”(.)“	”</p><p class="text">“
	or
	&rdquo;</p><p>&ldquo;
	or
	”</p><p>“

For single main quotes. Run manually (since some conversations do not have linebreaks):

Find:	Replace with:
&rsquo;(.)&lsquo;	&rsquo;</p><p class="text">&lsquo;
or	or
’(.)‘	’</p><p class="text">‘

	or &rsquo;</p><p>&lsquo; or ’</p><p>‘

Removing <sup> and <sub> tags (BV & CV)

ABBYY may add some <sup> or <sub> tags, lifting up words and letters for seemingly no reason. If you have no legitimate <sup> tags in your book (i.e. haven’t used them for footnotes/links), you can remove all the <sup>s in one go, using Regex. If you have intentionally used <sup>, you will of course have to check the search results manually.

Find: <sup>(.*?)</sup> and then <sub>(.*?)</sub>	Replace with: \1 (or nothing)

Remove anchors (BV & CV)

- such as <a id="bookmark0"></a> in which the id changes (to “bookmark1”, “referenceX”, etc). If you remove the HTML ToC there will often still be targets (anchors) left in the text. Remove them this way:

Find: <a id=(.*?)></a>	Replace with: nothing

Remove hyperlinks (BV & CV)

- such as <a href="http://address.org">Text here</a>, when there are several links with different URLs and you want to keep the text (“Text here”)

Find: <a href=(.*?)>(.*?)</a>	Replace with: \2

Remove mistaken underlining (BV & CV)

Sometimes ABBYY will underline words such as “him” and “think” that may seem underlined to the OCR, especially with serif fonts (which are used for most novels).

Find: <span style="text-decoration:underline;">(.*?)</span>	Replace with: \1

“l” (ell) instead of exclamation mark “!” (BV & CV)

Since exclamation marks are usually at the end of a sentence, search for “l” with lower case letters before and a space or a right-side quotation mark after. There will be a lot of suggested searches since quotation marks can be differently coded, and the space may be an ordinary one, or a &nbsp; or a &#160; — and then there may be span tags...

Inside a paragraph: a sentence that should end with an exclamation mark, followed by a new sentence starting with an upper case letter. (The ^/ addition is so that the top of the HTML page is ignored.) This will find several words legitimately ending with l followed by, usually, a name, too. Replace manually by editing in Code View, or use “! \1”, always checking first that the word doesn’t legitimately end with l:

```
[l](\s[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]
[l](&nbsp;[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]
[l](&#160;[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]
```

At the end of a paragraph. You can replace with “!</p>”, provided that you have already searched for bad line breaks:

```
[l]</p>
```

At the end of utterances ending a paragraph. You can replace with “!&rdquo;</p>” (etc, according to the quotation mark coding used in the text):

```
[l]&rdquo;</p>
or
[l]&rsquo;</p>
or
[l]”</p>
or
[l]’</p>
```

If you wish to continue and search also for styled sentences that should end with an exclamation mark, add </span> (or </b> or </i>) after [l].

“P” instead of question mark “?” (BV & CV)

Same reasoning as for exclamation marks since question marks also usually appear at the end of sentences.

In a paragraph: sentence that should end with an question mark followed by a new sentence starting with an upper case letter — replace manually by editing in Code View, or use “?\1”, always checking first that the word doesn’t legitimately end with P:

```
[P]([\s][A-Z"])
[P]([&nbsp;][A-Z"])
[P]([&#160;][A-Z"])
```

At the end of a paragraph; you can replace with “?</p>”, provided that you have already searched for bad line breaks:

```
[P]</p>
```

At the end of utterances ending a paragraph; you can replace with “?&rdquo;</p>” (etc, according to the quotation mark coding used in the text).

```
[P]&rdquo;</p>
or
[P]&rsquo;</p>
or
[P]”
or
[P]’
```

If you wish to continue and search also for styled sentences that should end with an question mark, add </span> (or </b> or </i>) after [P].

Span class such as italic or bold applied to single letter (BV & CV)

May be OCR error, but may also be correct if it’s a one-letter word (“Are you suggesting that *I* did it?”), so replace manually and check PDF when necessary.

Find:	Replace with:
<span class="\w+">(.?)</span> or <b>(.?)</b> or <i>(.?)</i>	\1

For a different first line following headings (CV)

Run when you are sure that all the chapter headings are in place.

Find:	Replace with:
</h2>\s+<p> and/or	</h2><p class="firstline"> and/or

</h3>\s+<p>	</h3><p class="firstline">

For larger first letter in first line (CV)

There might be a span class, such as italics, at the beginning of the first line, so start with this search:

Find:	Replace with:
<p class="firstline"><span(.*?)>(.)(.*?)</span>	<p class="firstline"><span\1><span class="firstletter">\2</span>\3</span>

Then the “main” search:

Find:	Replace with:
<p class="firstline">([A-Z])	<p class="firstline"><span class="firstletter">\1</span>

If the first character is a quotation mark, the above searches will (intentionally) not find it; the “main” search would run havoc with them, causing errors. So, run this one next in order to enlarge both quotation mark and the first actual letter of the first line:

Find:	Replace with:
<p class="firstline">&ldquo;(.) or <p class="firstline">&lsquo;(.) or <p class="firstline"><span class="firstletter">“</span>(.) or <p class="firstline"><span class="firstletter">‘</span>(.)	<p class="firstline"><span class="firstletter">&ldquo;\1</span> or <p class="firstline"><span class="firstletter">&lsquo;\1</span> or <p class="firstline"><span class="firstletter">“\1</span> or <p class="firstline"><span class="firstletter">‘\1</span>

You could still have a first line that starts with a span tag and then a quotation mark. Device another search, or simply go through your chapters and see if there’s a missing large letter somewhere…

Replace span styles with CSS classes (CV)

...for italic and bold (among other things) — you can leave it be, or you can be like me, and want all the styling in the stylesheet.

If so, the common occurrences are easy to deal with in normal search:

Find:	Replace with:
<span style="font-style:italic;">	<span class="italic">

and

Find:	Replace with:
<span style="font-weight:bold;">	<span class="bold">

Do note, however, that the bold style is often mistakenly applied when ABBYY for unknown reasons has interpreted normal-weighted text as bold.

ABBYY will also commit atrocities such as:

```
<span style="font-weight:bold;">V. L</span> <span style="font-weight:bold;font-style:italic;">f</span> <span style="font-weight:bold;">FADING</span>
```

It is actually easier to deal with these span combinations by clicking the buttons for bold or italic text than coming up with search terms to catch them all, or to edit out the spans in Code View. You will sometimes need to click twice to get rid of bold and/or italic styles. Just check with the PDF what it should be, if it is not obvious.

Using Normal mode

“1” (one) or “l” (ell) instead of “I” (BV & CV)

You will find most of them by a normal search for “1” with one space before and one after, and “l” with one space before and one after. Add a search for “<p>1” and “<p>l”, and/or “<p class="text">1” and “<p class="text">l” (search Case Sensitive for the “l” here), and you will catch the majority.

“rn” instead of “m” and vice versa (BV & CV)

A few examples:

modern - modem

burn - bum

corner - comer

Search for each, or leave it for the proofreading — assuming that you are sure you will spot them then. Don’t click “Replace All” — that could, for example, make modems modern.

“lie” instead of “he” (BV & CV)

Search obvious, of course checking for a true lie in the text each time.

“Pie” or “Fie” instead of “He” or “She” (BV & CV)

Search obvious, of course checking for real Pies and Fies in the text each time.

The above is simply a list of some common OCR mistakes that cannot be corrected by running a spellcheck. These mistakes are words and letters that can appear in the book legitimately, or indeed because the OCR software got it wrong. There’s no telling which it is without looking at the context in which they appear.

Since the mistaken words and letters may be legitimate, you can’t run an automated Find and Replace All. That might introduce new problems, by (for example) turning true bums into burns… So, you have to check for them “manually”, searching and clicking “Replace” only if the word is truly wrong.

The list is quite short: more of an inspiration for you than an attempt to be complete. It really is impossible to create a complete list — I will run searches for known occurrences like the ones above, but will often have to take a break in proofreading because some other OCR mistake has turned up, and seems to be repeated so that a search will save time compared to correcting each instance separately.

Unwanted span styles not previously corrected (CV)

I usually run a last search for “<span style=” with no other specification to find all such instances that ABBYY may have introduced, and may switch to a more specific “Find & Replace All” if there is some common issue, before continuing the unspecific “<span style=” search.

Applying quotefix (CV)

For double main quotes, left side:

Find:	Replace with:

&ldquo;&lsquo;
or
&ldquo; &lsquo;
or
&ldquo;&nbsp;&lsquo;
	<span class="quotefix">&ldquo;&nbsp;</span>&lsquo;

For double main quotes, right side:

Find:	Replace with:

&rsquo;&rdquo;
or
&rsquo; &rdquo;
or
	<span class="quotefix">&rsquo;&nbsp;</span>&rdquo;

&rsquo;&nbsp;&rdquo;	

For single main quotes, left side:

Find:
&lsquo;&ldquo;
or
&lsquo; &ldquo;
or
&lsquo;&nbsp;&ldquo;
	Replace with:
	<span class="quotefix">&lsquo;&nbsp;</span>&ldquo;

For single main quotes, right side:

Find:
&rdquo;&rsquo;
or
&rdquo; &rsquo;
or
&rdquo;&nbsp;&rsquo;
	<span class="quotefix">&rdquo;&nbsp;</span>&rsquo;

The exception: Flatten ToC with Calibre e-book editor (BV & CV)

I have come across EPUBs with ToCs that were levelled when they shouldn’t be a few times. ToCs are legitimately levelled when there are, for example, a chapter h2 heading and a number of h3 (sub-)headings in that chapter. The ToC will show the h3 headings one level down, under the h2 heading.

Then there are those ToCs in which, for example, a h2 is inexplicably placed “under” another h2. It can’t be fixed by correcting the headings, because they are already correct.

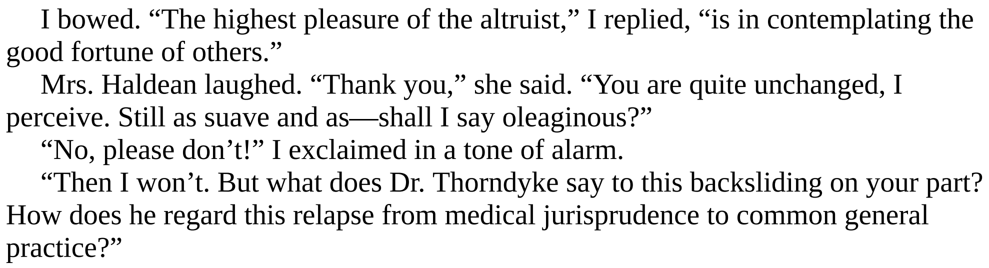
This is when the Calibre e-book editor comes in handy — at least if there are no legitimately levelled headings in the book.

After opening the EPUB in the editor, go to Tools → “ToC - Edit Toc”, and then press the “Flatten ToC” button. Save and close the program.

The Finishing Touches

Add your custom CSS (BV)

This is optional for the Book View editor: use a stylesheet for applying styles that are not available in the GUI, notably for headings and paragraphs (text-indent for running text in a novel). Use the “Replacement CSS” from the Stylesheets chapter here (or create your own) if you want to have control of the headings, and have running text without blank spaces between the paragraphs and indented like this:



> I bowed. “The highest pleasure of the altruist,” I replied, “is in contemplating the good fortune of others.”
> Mrs. Haldean laughed. “Thank you,” she said. “You are quite unchanged, I perceive. Still as suave and as—shall I say oleaginous?”
> “No, please don’t!” I exclaimed in a tone of alarm.
> “Then I won’t. But what does Dr. Thorndyke say to this backsliding on your part? How does he regard this relapse from medical jurisprudence to common general practice?”

Open the “Styles” folder to the left in Sigil and double-click “main.css” to open it. Copy the text of the “Replacement CSS”, and paste it into the css file. N.B. if you left classes with italics in the style sheet before, don’t remove them now. Paste the “Replacement CSS” in addition to such classes. Save (File → “Save”, or Ctrl+S, just like any word processor).

If you “only” want indented text, but prefer the headings Sigil gave you, remove the h-lines in the stylesheet.

Clean up the stylesheet (BV & CV)

When the book is finished, I run Tools → “Remove Unused Stylesheet Classes”, which makes any unused special class disappear. It doesn’t happen right away: you will get a preview of what is about to be deleted. Standard classes, such as h2, h3, and td will remain even if they are not used in the HTML pages, so you need to remove them manually from the style page if you want it perfectly slimmed down. Provided that you really did not use them, of course.

Metadata

If you converted your PDF from ABBY, you have probably already added Title and Author to the EPUB's metadata by entering them as you saved. If your EPUB lacks those entires, or if you want to edit them, got to Tools → "Metadata Editor...". You want your book to have at least those two pieces of metadata.

Validate with Flight Crew (BV & CV)

Normally there will be nothing to correct here — but if the check shows some error, you will of course have to fix it. Flight Crew checks that the EPUB is coded according to the XHTML standard, structured as it should be, and contains everything it should (in addition to the text itself), so errors might mean that your EPUB will not work as intended when you try to read on your e-reader.

Proofreading (BV & CV)

Yes, it takes time and is often boring, compared to just reading, but it is worth it for the quality gain. There are usually lots of little OCR mistakes not previously detected.

I used to proofread using my e-reader, getting up all the time for performing corrections in Sigil on the computer. *That* was boring!

Then I got myself a small laptop, installed Sigil, and have since proofread using that one. This is one place where I really want to keep using the "old" Sigil, with the ability to easily switch between Book View and Code View. I copy the searchable PDF to the laptop, too, so that I can quickly check if I am unsure about whether something really is correct or not.

One thing to look out for, apart from all the possible ordinary issues: ABBYY occasionally discards whole chunks of text, typically the paragraph(s) that follow a heading, an image, or a header. Most of the time you catch it when searching for bad line breaks. But without carefully reading the EPUB against the PDF, there is no way to be sure that the EPUB really contains all the text of the original…

I am not enough of a perfectionist to do that comparative reading, and I suspect that none or few others are, either. But: if you notice a "jump" in the text, do go back to the PDF and check if there really is something missing.

Check with e-reader (BV & CV)

One final step before you can sit back and be really pleased with what you have achieved…

Load your EPUB into your e-reader and make sure that you like what you see. Also, test it with one or more e-readers on your computer.

And then I say: **WELL DONE!** Welcome to the club of proud creators of quality EPUBs!

Appendix: Searches for Easy Copying

This is a list of the correction/fix searches for easy copying - note that some of them may wreck your EPUB if run with the "Replace All" option, so you do need to read the comments in the Corrections and Fixes chapter before use. Also, the numbering below may refer to alternatives or order to run, so you need to check up on that, too.

```
REGEX:

Missing chapter heading tags 1
Find:
<p>Chapter (.*?)</p>
Replace with:
<h2>Chapter \1</h2>


Missing chapter heading tags 2
Find:
<body>\s+<p class="(.*?)">(\d+)</p>
Replace with:
<body><h2>\2</h2>


Missing chapter heading tags 3
Find:
<body>\s+<p>(\d+)</p>
Replace with:
<body><h2>\1</h2>


Page numbers remaining after OCR 1
Find:
<p class="(.*?)">Page \d+</p>
Replace with:
nothing


Page numbers remaining after OCR 2
```

```
Find:
<p class="(.*?)">\d+</p>
Replace with:
nothing
```

```
Page numbers remaining after OCR 3
Find:
<p>Page \d+</p>
Replace with:
nothing
```

```
Page numbers remaining after OCR 4
Find:
<p>\d+</p>
Replace with:
nothing
```

```
Bad line breaks 1a
Find:
</p>\s+<p class="(.*?)">([a-z]\w*\s)
Replace with:
 \2
N.B. the space before \2
```

```
Bad line breaks 1b
Find:
</p>\s+<p>([a-z]\w*\s)
Replace with:
 \1
N.B. the space before \1
```

```
Bad line breaks 2a
Find:
([^.?!;:>])</p>\s+<p class="(.*?)">
Replace with:
\1
N.B. the space after \1
```

```
Bad line breaks 2b
Find:
([^.?!;:>])</p>\s+<p>
Replace with:
\1 
N.B. the space after \1


Missing line breaks in conversations - dq 1
Find:
&rdquo;(.)&ldquo;
Replace with:
&rdquo;</p><p class="text">&ldquo;
or
&rdquo;</p><p>&ldquo;

Missing line breaks in conversations - dq 2
Find:
”(.)“
Replace with:
”</p><p class="text">“
or
”</p><p>“


Missing line breaks in conversations - sq 1
Find:
&rsquo;(.)&lsquo;
Replace with:
&rsquo;</p><p class="text">&lsquo;
or
&rsquo;</p><p>&lsquo;


Missing line breaks in conversations - sq 2
Find:
’(.)‘
Replace with:
’</p><p class="text">‘
or
```

```
’</p><p>‘


Removing <sup> tags
Find:
<sup>(.*?)</sup>
Replace with:
\1 (or nothing)

Removing <sub> tags
Find:
<sub>(.*?)</sub>
Replace with:
\1 (or nothing)


Remove anchors
Find:
<a id=(.*?)></a>
Replace with:
nothing (blank "Replace" box)


Remove hyperlinks
Find:
<a href=(.*?)>(.*?)</a>
Replace with:
\2


Mistaken underlining
Find:
<span style="text-decoration:underline;">(.*?)</span>
Replace with:
\1


ell instead of exclamation mark 1
Find:
[l](\s[A-Z]\w*)[^/(?!PUBLIC|xmlns|version)]
!\1
```

```
ell instead of exclamation mark 2
Find:
[l]</p>
Replace with:
!</p>

ell instead of exclamation mark dq 1
Find:
[l]&rdquo;</p>
Replace with:
!&rdquo;</p>

ell instead of exclamation mark dq 2
Find:
[l]”</p>
Replace with:
!"</p>

ell instead of exclamation mark sq 1
Find:
[l]&rsquo;</p>
Replace with:
!&rsquo;</p>

ell instead of exclamation mark sq 2
Find:
[l]’</p>
Replace with:
!’</p>


P instead of question mark 1
Find:
[P]([\s][A-Z"])
Replace with:
?\1


P instead of question mark 2
Find:
[P]</p>
Replace with:
```

```
?</p>


P instead of question mark dq 1
Find:
[P]&rdquo;</p>
Replace with:
?&rdquo;</p>


P instead of question mark dq 2
Find:
[P]”
Replace with:
?”</p>


P instead of question mark sq 1
Find:
[P]&rsquo;</p>
Replace with:
?&rsquo;</p>


P instead of question mark sq 1
Find:
[P]’
Replace with:
?’</p>


Span class applied to single letter
Find:
<span class="\w+">(.?)</span>
Replace with:
\1


Bold tag applied to single letter
<b>(.?)</b>
Replace with:
\1
```

```
Italic tag applied to single letter
<i>(.?)</i>
Replace with:
\1


For a different first line following h2 headings
Find:
</h2>\s+<p>
Replace with:
</h2><p class="firstline">


For a different first line following h3 headings
</h3>\s+<p>
Replace with:
</h3><p class="firstline">


For larger first letter in first line with span
Find:
<p class="firstline"><span(.*?)>(.)(.*?)</span>
Replace with:
<p class="firstline"><span\1><span class="firstletter">\2<


For larger first letter in plain first line
Find:
<p class="firstline">([A-Z])
Replace with:
<p class="firstline"><span class="firstletter">\1</span>


For larger first letter in first line dq 1
Find:
<p class="firstline">&ldquo;(.)
Replace with:
<p class="firstline"><span class="firstletter">&ldquo;\1</
```

For larger first letter in first line dq 2

Find:

```
<p class="firstline"><span class="firstletter">“</span>(.)
```

Replace with:

```
<p class="firstline"><span class="firstletter">“\1</span>
```

For larger first letter in first line sq 1

Find:

```
<p class="firstline">&lsquo;(.)
```

Replace with:

```
<p class="firstline"><span class="firstletter">&lsquo;\1</
```

For larger first letter in first line sq 2

Find:

```
<p class="firstline">‘</span>(.)
```

Replace with:

```
<p class="firstline"><span class="firstletter">‘\1</span>
```

Replace span style italic with class

Find:

```
<span style="font-style:italic;">
```

Replace with:

```
<span class="italic">
```

Replace span style bold with class

Find:

```
<span style="font-weight:bold;">
```

Replace with:

```
<span class="bold">
```

NORMAL MODE:

Search for bum for burn, etc.
Search for lie for he, if it seems to happen a lot in the
Search for Pie or Fie for He or She if it seems to happen

one instead of I - no. 1

```
Find:
 1
Replace with:
 I
N.B. spaces before and after


one instead of I - no. 2
Find:
<p class="text">1
Replace with:
<p class="text">I


one instead of I - no. 3
Find:
<p>1
Replace with:
<p>I


ell instead of I - no. 1
Find:
 l
Replace with:
 I
N.B. spaces before and after


CASE SENSITIVE:
ell instead of I - no. 2
Find:
<p class="text">l
Replace with:
<p class="text">I


CASE SENSITIVE:
ell instead of I - no. 3
Find:
<p>1
Replace with:
```

```
<p>I


Double main quotefix, left side
Find:
&ldquo;&lsquo;
or
&ldquo; &lsquo;
or
&ldquo;&nbsp;&lsquo;
Replace with:
<span class="quotefix">&ldquo;&nbsp;</span>&lsquo;


Double main quotefix, right side
Find:
&rsquo;&rdquo;
or
&rsquo; &rdquo;
or
&rsquo;&nbsp;&rdquo;
Replace with:
<span class="quotefix">&rsquo;&nbsp;</span>&rdquo;


Single main quotefix, left side:
Find:
&lsquo;&ldquo;
or
&lsquo; &ldquo;
or
&lsquo;&nbsp;&ldquo;
Replace with:
<span class="quotefix">&lsquo;&nbsp;</span>&ldquo;


Single main quotefix, right side:
Find:
&rdquo;&rsquo;
or
&rdquo; &rsquo;
or
```

```
&rdquo;&nbsp;&rsquo;
Replace with:
<span class="quotefix">&rdquo;&nbsp;</span>&rsquo;
```